

WO 02/37777 A1



(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

LF, MA, MD, MG, MK, MN, MH, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:

— as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for the following designations AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU,

Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS ENABLING STANDARD VOICE TELEPHONE
TO INITIATE AND RECEIVE VOICE TELEPHONE CALLS ON
TELEPHONE LINE OCCUPIED WITH DIAL-UP INTERNET CONNECTION

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention concerns the field of telecommunications and provision of voice calls over the Internet. More particularly, the invention enables the user of a standard telephone to make and receive voice telephone calls over one telephone line even while this line is already occupied by a personal computer (PC) connected to an Internet Service Provider (ISP). The operation of the telephone is unchanged, and the PC user can still browse the world wide web, upload and download files, and perform other such activities without affecting the quality of the voice telephone call.

2. Description of the Related Art

For households with only one telephone line, being connected to the Internet means that this telephone line is unavailable to make and receive normal telephone calls. Technically speaking, the telephone line is connected to an ISP via the public switched telephone network (PSTN). This connection occupies the telephone line to the exclusion of any normal, voice telephone calls. The call waiting feature must be deactivated to prevent errors from occurring in the Internet connection when incoming calls are received. Thus, the telephone's existing coupling to the PSTN prevents occupants of the household from placing or receiving new telephone calls. Since the average duration of a personal Internet connection is about thirty-five minutes, people wishing to make a normal voice telephone call can experience lengthy delays.

A number of different approaches have been developed in an attempt to address this concern. One example is Digital Subscriber Line (DSL) service, in which one

telephone line is shared by a voice telephone call and a high speed Internet connection. This is accomplished by allocating one frequency band for voice information and a different band for Internet data. Although DSL service benefits customers in various ways, there are drawbacks. Chiefly, DSL services often come with substantial costs. Use of DSL services requires purchase of a modem and Ethernet card, payment for installation services, and payment of repeating monthly access fees. In some markets, the price that local carriers charge for provision of this service is too high to encourage widespread acceptance.

Another approach to simultaneously conducting Internet and voice telephone data on a single telephone line works by superimposing voice on an existing Internet connection using "Voice-Over-Internet-Protocol" (VoIP) technology. One implementation of VoIP enables customers to use technology computers to conduct voice telephone calls with other users via their computer. This computer-to-computer approach requires both computer users to install proprietary VoIP software, such as the NETMEETING program of MICROSOFT CORPORATION. Initially, each computer user logs on to the Internet by dialing an ISP. Then, each user directs his/her computer to exchange information with the other computer, thereby forming a connection over the Internet between the computers. Each computer requires its user to speak and listen to the other user by using a sound card and microphone, respectively.

A different implementation of VoIP technology is the Ericsson IPT voice gateway and associated Phone Doubler software. The Ericsson IPT voice gateway, which resides at a central facility such as an ISP, provides hardware and associated software for originating and terminating ISP customers' VoIP calls. Initially, a VoIP customer installs

Ericsson's Phone Doubler software on his/her computer. The phone doubler software is a client server application that is designed to be resident on a computer, and acts to set up session with the Ericsson IPT voice gateway. When the VoIP customer wishes to place a call, he/she uses his/her computer to dial the called party, either as a standard PSTN phone number or an alias that is known to the Ericsson IPT voice gateway to represent a destination either on the PSTN or the Internet. The called party may be another VoIP customer, or a person with a standard PSTN telephone. The Ericsson IPT voice gateway establishes the connection to the called party. Calls may also be initiated in the reverse direction. Namely, the VoIP customer can receive calls initiated by another VoIP or non-VoIP party. In any case, the VoIP customer must use his/her computer to originate and receive voice telephone calls when the customer's line is occupied by an Internet connection. Namely, the VoIP customer is required to use the PC computer's microphone and sound output card to complete the conversation, and the computer translates between analog voice signals and VoIP format.

One drawback of the foregoing VoIP solutions is that customers must learn how to utilize a computer microphone and speaker instead of the standard telephone handset. For some users, this may be undesirable because of the additional training required, and because of people's penchant for the familiar feel of a telephone handset. Another drawback is that computers are relatively immobile, especially compared to cordless telephones. Furthermore, some users may not be satisfied with the quality of conventional VoIP telephone calls due to the limited fidelity of many sound output cards, computer microphones, and computer speakers. Furthermore, computer users may be frustrated because such applications occupy the computer's sound output card to the exclusion of

any other PC programs, preventing the full use of computer games, music players, teaching programs, and other applications that utilize sound.

As one alternative, to enable the receipt of inbound phone calls, some commercially available modems offer the feature of briefly interrupting an Internet connection to notify the user of an inbound call. However, to take the call the connection to the Internet must be broken. Such devices do not allow for the making of a phone call while the Internet is in use. A different approach is exemplified by WIPO document WO 97/47127, published on December 11, 1997. This approach concerns a modem with IP support for simultaneous access to a switched telephony network and to the services of an IP based network. Although this approach might be satisfactory for certain applications, some users could find it too expensive for their budget.

Consequently, known approaches to sharing telephone connections between voice conversations and Internet data are not completely adequate due to certain unsolved problems.

SUMMARY OF THE INVENTION

By synergistically combining existing PC software, new PC software, and a new Internet Home Phone (IHP) device, the invention enables a customer to use a standard telephone to place and receive telephone calls over one telephone line even when this line is already occupied by a connection between the customer's PC and the Internet. Advantageously, the operation of the standard telephone unchanged. Additionally, the customer can simultaneously use the PC to browse the worldwide web, upload and/or download files, and perform other Internet operations without affecting the quality of the

voice telephone call.

More specifically, the invention utilizes an IHP device, which is attached to a standard telephone handset, a PC, and a telephone line. Whenever the PC is not connected to the Internet, the IHP device connects the handset to the telephone line to initiate and receive calls normally. When the PC is connected to the Internet, and the user initiates a call from the handset, the IHP device provides a simulated dial tone back to the handset. The IHP device also captures the number dialed on the handset, converts it to a PC compatible format, and transmits the dialed number to the PC. Software on the PC receives the dialed number from the IHP device, and checks to ensure that a valid phone number has been dialed. Once the PC receives a valid phone number, it transmits the dialed number to a voice gateway via switching in the IHP device. The voice gateway responds by routing the call appropriately to establish a connection to the called party. If the called party is not using VoIP, the voice gateway translates between PSTN-compatible format (for the called party) and VoIP format (for the IHP customer). When the called party answers, the voice service is established. The PC and the IHP device work together with the voice gateway to maintain a two-way VoIP conversation.

On the other hand, when the PC is connected to the Internet and an incoming call occurs, provided that a call diversion to the voice gateway is in place, there is no need to automatically return a busy signal to the caller. Instead, the local PSTN diverts calls back to the voice gateway, which routes the call under IP format via the ISP to the PC. In response, the IHP customer's PC software sends a signal to the IHP device causing the IHP device to ring the telephone handset. When the customer answers the call using the telephone handset, the IHP device and PC conduct a two-way VoIP conversation as

described above. As an alternative, the user may answer incoming calls using PC hardware such as a sound output card and a microphone.

During a call, whether incoming or outgoing, the IHP device receives speech from the telephone handset and converts it into digital signals. The IHP device may also compress these digital signals in order to more efficiently use the limited bandwidth between the IHP device and the PC. Software on the PC encapsulates digital signals received from the IHP device into Internet compatible packets for transmission via the Internet to the voice gateway. Conversely, PC software also extracts digitized voice signals from Internet packets received from the voice gateway and transmits the digitized voice signals to the IHP device. The IHP device converts the digitized voice signals into an appropriate analog format for transmission to the telephone handset's ear piece. When the conversation ends by either party hanging up, the IHP device and PC reset to await a new call. The voice gateway then creates a call record that details the calling party, the called party, and the duration of the call for billing purposes. Thus, the foregoing hardware and software components enable the IHP customer to conduct simultaneously Internet and voice calls over one telephone line.

The foregoing features may be implemented in a number of different forms. For example, the invention may be implemented to provide a method of interfacing a standard telephone with a telephone line that is already occupied by a PC-to-Internet connection, thereby enabling the user of the telephone to make and receive telephone calls despite the existing Internet connection. In another embodiment, the invention may be implemented to provide an apparatus comprising an IHP device enabling telephone callers to place voice telephone calls over a telephone line that is already occupied by an existing Internet

connection. In still another embodiment, the invention may be implemented to provide a signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital data processing apparatus, such as the PC and/or IHP device, to provide telephone/Internet interfacing as described herein. Another embodiment concerns logic circuitry having multiple interconnected electrically conductive elements configured to provide such telephone/Internet interfacing.

----- The invention affords its users with a number of distinct advantages. For example, the invention enables telephone callers and call recipients to enjoy the familiar comfort of their normal telephone handsets during VoIP telephone calls. As another advantage, callers can enjoy less expensive long distance and international calls through their ISP rather than traditional long distance networks and calling plans. Along these lines, the invention helps telephone customers save costs by utilizing one telephone line to conduct voice calls and Internet connections, without purchasing a second telephone line. As still another advantage, the present invention's telephone solution interfaces seamlessly with a standard telephone handset, giving similar quality of connection as enjoyed when making a standard phone call. Another additional benefit of the invention is its automated nature, including automatic reversion to normal telephone handset functions in the event of power loss, and automatic operation of the traditional telephone handset as an Internet phone when a call is initiated or received on a line with an existing Internet connection. Additionally, the present invention reduces costs by utilizing existing software and processing capability of the user's PC. The invention also provides a number of other advantages and benefits, which should be apparent from the following description of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURES 1A-1B show block diagrams of the hardware components and interconnections of an Internet access system according to the invention.

FIGURE 2 is a block diagram of a digital data processing machine according to the invention.

FIGURE 3 shows an exemplary signal-bearing medium according to the invention.

FIGURE 4 is a flowchart showing the overall operating sequence of the invention.

FIGURE 5 is a flowchart depicting an initial log-on sequence, according to the invention.

FIGURES 6A-6C show a flowchart depicting an outgoing call sequence, according to the invention.

FIGURE 7 is a flowchart depicting conversion of voice signals from the telephone handset microphone to IP format, according to the invention.

FIGURE 8 is a flowchart depicting conversion of an incoming signal from IP format to an analog voice signal suitable for transmission to the telephone handset earpiece, according to the invention.

FIGURE 9 is a flowchart depicting a sequence for call termination by a remote party, according to the invention.

FIGURE 10 is a flowchart depicting a sequence for call termination by the IHP customer, according to the invention.

FIGURE 11 is a flowchart depicting an incoming call sequence, according to the invention.

DETAILED DESCRIPTION

The nature, objectives, and advantages of the invention will become more apparent to those skilled in the art after considering the following detailed description in connection with the accompanying drawings.

HARDWARE COMPONENTS & INTERCONNECTIONS

Overview

Broadly, this invention utilizes commercially available PC hardware, new PC software, a new IHP device, and commercially available telephony services to enable customers to use a standard telephone to make and receive telephone calls even when their PCs are connected to the Internet. Not only is the operation of the phone unchanged, but the user can simultaneously use the PC to browse the worldwide web, upload and/or download files, and the like without affecting the quality of the voice telephone call.

FIGURES 1A-1B depict a system 100 that employs features of this invention. To illustrate the invention in context, the system 100 includes some commercially available components along with other components newly added by this invention. The system 100 includes a PC 102, IHP device 106, telephone handset 108, and various remote telephone facilities 182. Broadly, the PC 102 and IHP device 106 include inventive software programming and hardware enabling people to conduct a telephone conversation (using the handset 108) while simultaneously conducting an Internet session (using the PC 102). The remote telephone facilities 182 include one or more voice gateways, public switched telephone networks (PSTNs), and at least one ISP.

Remote Telephone Facilities 182

Introduction

The remote telephone facilities 182 are located remotely from the customer, and serve to appropriately route incoming and outgoing calls between one customer and another, or between one customer and the Internet. In the specific context of this invention, the remote facilities 182 carry VoIP and Internet data calls between the IHP device 106 and a remote call-receiving or call-originating party, as the case may be.

The remote telephone facilities 182 include one or more PSTNs, ISPs, and voice gateways, as well as various VoIP and non-VoIP service provider IP infrastructure. These components may be implemented by commercially available hardware devices. In order to implement the invention, the voice gateway 146a is programmed to recognize the IHP customer. Optionally, the ISP 147a may be reconfigured in certain ways to route VoIP traffic more effectively. As another option the PSTN 149a may be configured to forward the IHP customer's incoming calls to the voice gateway 146a. To provide a tangible example, FIGURE 1B illustrates the remote telephone facilities 182 in the context of various exemplary call routes 195-197.

Call Route 195

The call route 195 connects the IHP device 106 (FIGURE 1A) to a desired "called party," such as another PC and/or a standard telephone handset. In the case of normally initiated telephone calls of the IHP customer, these calls proceed to the PSTN 149a and then to non-VoIP infrastructure 180 such as the same or other PSTNs, long distance networks, etc. Outgoing Internet calls from the IHP customer progress through the PSTN 149a, ISP 147a, and voice gateway 146a. The voice gateway is operated by a "VoIP

service provider." Depending upon the circumstances, the voice gateway 146a can route the call via non-VoIP infrastructure 180 and the PSTN 149b, or via VoIP service provider IP infrastructure 155 and alternative voice gateway 146b or 146c, etc.

When the IHP customer is not logged in to the Internet, incoming voice calls arrive from the PSTN 149a via the non-VoIP infrastructure 180 (such as the same or another PSTN, long distance network, etc.) When the IHP customer is logged in to the Internet, incoming voice calls arrive according to various routings. According to one routing, if the call originates from a standard telephone connected to a PSTN it is routed by the call originator's carrier to the IHP customer's local carrier supplying PSTN 149a. Then a call diversion command redirects the inbound call to the voice gateway 146a. The call diversion command is initiated by the IHP customer to the local carrier that supplies PSTN 149a prior to establishing the connection to their ISP 147a to log on to the Internet. The call forward command to the local carrier is a dialed sequence which may vary between local carriers, and may constitute a sequence of "72*" followed by the telephone number of the voice gateway 146a. Once the call has been delivered by the PSTN 149a to the voice gateway 146a, the voice gateway converts the call to a VoIP call and routes the call via the affiliated ISP 147a and the PSTN 149a to the PC 102 which is connected to the IHP device 106 that routes the call to the IHP customer's telephone handset 108.

In the other option, if the calling party is a customer of the VoIP service provider and is connected to one of VoIP service provider's gateways, the call is routed across the IP infrastructure 155 on a least cost basis to voice gateway 146a which will then route the call to IHP device 106 as described above.

Call Route 196

The call route 196 connects a standard telephone handset 177 to the non-VoIP infrastructure 180, for the purpose of initiating a call to the IHP customer or receiving a call from the IHP customer. The call route 196 includes the PSTN 149c, which may be the same as the PSTN 149a, 149b for customers located in the same area. The PSTN 149c is coupled to the non-VoIP infrastructure 180, and may also be attached to a voice gateway 146c accessible by the VoIP provider IP infrastructure 155. The call route 196 may operate under conventional principles of telephony.

Call Route 197

The call route 197 connects the IHP customer to another VoIP customer, utilizing a PC 165 (for PC-to-PC VoIP communication) or IHP device (for VoIP communication as discussed herein). Similar to the call route 195, the call route 197 includes a PSTN 149d, VoIP compatible ISP 147b, voice gateway 146b, and PSTN 149e coupled to the non-VoIP infrastructure 180.

Details

As used herein, the PSTNs 149a-149e represent one or more local call routing facilities, which may be exemplified by Regional Bell Operating Company (RBOCs). The PSTNs serve as a communications link between customers' telephone lines to long distance networks, ISPs, routers, VoIP facilities, and other telephone facilities.

The ISPs 147a-147b include routing equipment, switching equipment, and telecommunication line access necessary to provide Internet access to ISP customers.

Depending upon the application, the ISP 147a may be associated with telephone companies (such as PSTN companies), or not. To provide enhanced VoIP compatibility as described herein, the ISPs 147a-147b may optionally include additional hardware and/or software appropriate to optimize the performance of their network for the transmission of voice traffic. Some examples include: (1) voice packet prioritization across the ISP's infrastructure including but not limited to Remote Access Servers (RASs), Routers, switches etc., and (2) IP packet length optimization. Setting up all the infrastructure used by the ISP 147a in the signal path between the point of ingress to the ISP network IHP customer to the point of egress to the voice gateway 146a to limit the maximum length of an IP packet. For example, the maximum length of an IP packet may be limited to 400 bytes to ensure that the delay in transmitting the voice packet over the local PSTN 149a is limited to 10 milliseconds.

Broadly, each of the voice gateways 146a-146c comprises equipment that functions to (1) selectively provide the IHP customer with access to the services of that voice gateway, (2) manage the transmission of VoIP traffic across the VoIP service provider infrastructure 155 to ensure that the least cost route is used to meet the performance requirements of the IHP customer, (3) convert the VoIP signals into signals compatible with a standard PSTN compatible voice call and vice versa, (where applicable) and (4) attend to billing of the IHP customer for VoIP services. The gateway 146 may also perform other functions such as call authentication, billing, automated voice response based services, advanced services such as audio conferencing, etc. An exemplary hardware component is the Ericsson model IPT voice gateway product. The voice gateways 146a-146b are

individually associated with the respective ISPs 147a-147b, and operate in cooperation therewith.

The VoIP service provider infrastructure 155 includes hardware such as routers, switches, servers and transmission capacity between the voice gateways locations. The transmission capacity may be either IP based, clear channel, ATM, SONET, dark fibre, microwave or other telecommunications services that may be available to link two location. Broadly, this hardware performs the function of transmitting the VoIP calls between the voice gateways, manages the transmission of this traffic to ensure that the quality of the voice call is not adversely effected.

The non-VoIP infrastructure 180 includes all other telecommunications infrastructure required to deliver the phone call to the called party. For example, the infrastructure 180 may include long distance services, international phone services, mobile phone services, inter-exchange carrier services, local exchange carrier services, etc.

Telephone Handset

Referring to FIGURE 1A, one particular advantage of this invention is that a caller can use a familiar telephone handset to place and receive voice telephone calls while his/her telephone line is already occupied by an Internet connection. Namely, the handset 108 is located at the customer's premises, and comprises a traditional land line telephone handset such as a desktop phone, wall mount phone, cordless phone, etc. The handset 108 may be embodied by a unitary piece, or by a separate cradle and hand piece.

Personal Computer (PC)

The PC 102 comprises a general purpose computer reconfigurable by the IHP customer to run a variety of different programs. Accordingly, the invention is able to take advantage of certain processing power already present in the user's PC to supplement the IHP device 106. In the illustrated embodiment, invention utilizes commercially available software existing on the PC as well as some new components. The PC 102 may comprise a conventional desktop computer, notebook computer, computer workstation, or other appropriate digital data processing machine. In the illustrated example, the PC 102 comprises a personal computer from a manufacturer such as IBM, DELL, or COMPAQ, where this PC 102 utilizes an operating system such as a MICROSOFT WINDOWS operating system. As ordinarily skilled artisans (having the benefit of this disclosure) will recognize, other operating systems may be used instead, such as Unix, Linux, etc.

The PC 102 includes a CPU 111 and ports 116, 114. The CPU 111 includes various software sub-components, comprising IHP client software 110, VoIP engine 113, sound signal router 118, and IP manager 112. In the ill The PC 102 may also include other software normally present in PCs.

PC Hardware

In the present example, the ports 116, 114, comprise RS232 or USB type ports, although other communications ports may be employed such as Local Area Networking (LAN) ports, infrared coupling ports, SCSI ports, parallel ports, etc. In an alternative embodiment, where the IHP device 106 is manufactured as a PC card installed in the PC 102, then the ports 116, 114 are implemented by internal structure of the PC 102 such as busses, backplanes, cables, mother boards, etc.

The port 116 is coupled to the IHP device 106 by a cable 117 such as a serial, USB, or other cable appropriate to the port. The port 116 communicates with the IHP device 106 in order to conduct digitally encoded voice, instructions, and status messages between the PC 102 via the IHP device 106. The port 116 conducts communications directly with the IHP client software 110, as well as communications with the VoIP engine via the sound signal router 118.

The port 114 is coupled to a telephone modem 104 for the purpose of exchanging VoIP data and Internet session data with the remote telephone facilities 182 via the IHP device 106. An exemplary modem 104 is provided by a Hayes Accura brand 56K modem. The modem 104 may be provided as a component internal to the IHP device 106 (as shown), incorporated into the programming of the processor 130, internal to the PC 102, or a standalone component between the PC 102 and IHP device 106. Integrating the modem 104 into the IHP device 106 may be desirable from the standpoint of reducing complexity of installation, and improving quality of VoIP service. Additionally, in embodiments where the IHP device 106 is manufactured as a PC card, this card may be sold as an alternative to currently available modem cards. In this embodiment, the IHP device 106 (from the outside) may provide a similar appearance as today's familiar modem cards, which provide "line" and "telephone" ports.

In addition to the foregoing components, the PC 102 includes other hardware components such as a processor, data storage, input/output, and other components as specifically discussed below.

PC Software - Introduction

One function of the PC software is to manage an Internet connection, which may occur in the same manner as other conventional PC-initiated Internet connections. Additionally, however, the PC software performs various additional functions, namely taking appropriate action when the telephone handset 108 is used to place a call, or an incoming voice call arrives, while an Internet data connection is already in place. In such event, the PC software employs the existing Internet data connection to relay VoIP data.

PC software - VoIP Engine

The VoIP engine 113 comprises a software module that sets up session with the voice gateway 146a. As one example, the VoIP engine 113 may comprise an ActiveX component containing all VoIP functionality required to make and receive VoIP calls. One example of this is the VoIP engine provided with the Ericsson Phone Doubler software. Although other VoIP engines may be commercially purchased or developed anew, the present disclosure utilizes the VoIP engine 113 as being implemented by the Ericsson VoIP engine. The VoIP engine of Ericsson's Phone Doubler software includes programming to make and receive calls using TCP/IP connections (on the Internet side) and a sound card (for the user side). The Ericsson Phone Doubler software also includes a supervisory client program to control the VoIP actions, which is extracted from the VoIP engine and discarded in the presently illustrated implementation of the invention. Instead, as discussed below, the present invention utilizes unique client software 110 to manage the VoIP engine 113, among other tasks.

PC Software - IHP Client Software

The IHP client software 110 provides a number of functions. Chiefly, the software 110 manages the VoIP engine 113, and if desired, may provide a GUI or other interface to the VoIP engine 113. For example, to set up a VoIP call, the software 110 sends appropriate commands to the VoIP engine 113, which then establishes a call session with a voice gateway. This includes communicating with an authorization server to establish billing information, provide passwords, supply account IDs, etc.

Additionally, when first activated by the IHP customer, the software 110 communicates with the voice gateway service provider to register the IHP customer and configure voice gateway services for the customer. Later, the client software 110 serves as a buffer to collect incoming telephone numbers dialed by the handset 108 and diverted to the PC 102 by the IHP device 106 because the line 145 is already coupled to the Internet. In addition, when the IHP customer instructs the PC 102 to connect to the Internet by activating the software 110, the software 110 directs the VoIP engine 113 to establish a connection to the voice gateway 146a.

Also, to support calls initiated from the handset 108 via the IHP device 106, the client software 110 receives information from the IHP 106 as to actions occurring at the handset 108, e.g., off hook, DTMF tone dialed, on hook etc. For example, when the client software 110 detects the telephone off hook, it directs the IHP device 106 to generate a dial tone. DTMF digits are received from the IHP 106 as the user dials a number. Using a table of allowable number sequences, the client software 110 decides when a valid phone number has been entered. At this point, the client software 110 directs the VoIP engine 113 to establish a VoIP call to the voice gateway. The client software 110 also

interacts with the sound signal router 118 in order to send commands to the IHP and to receive status information.

PC Software - Sound Signal Router

Broadly, the sound signal router 118 directs digitized sound signals between the VoIP engine 113 and the IHP device 106. In the presently illustrated embodiment, utilizing a VoIP engine 113 such as Ericsson's Phone Doubler software designed to work with a computer sound card, the router 118 performs the task of redirecting sound input/output to the IHP device 106 rather than the sound card. As a result, the PC user is free to operate his/her sound card for games, listening to music, and other functions while conducting a VoIP call. More particularly, the router 118 routes digitized sound signals between the VoIP engine 113 and the IHP device 106 via the port 116.

In the MICROSOFT WINDOWS environment, the router 118 may be implemented by a dynamic link library (DLL). The VoIP engine 113 as implemented by the Ericsson Phone Doubler normally communicates with a PC sound card using a WINDOWS DLL, WINMM.DLL. All sound information on a PC system passes through this DLL to get from the user software to the sound hardware. The VoIP engine 113 calls WINMM.DLL in order to process the sound data. To enable the VoIP engine 113 to use the IHP device 106 in place of the sound card, the VoIP engine 113 and/or operating system is modified so that the VoIP engine 113 calls a new DLL entitled "NTMMM.DLL" instead of the usual WINMM.DLL. An exemplary listing of NTMMM.DLL is shown below in TABLE 1.

The NTMMM.DLL has a switch built into it so that the sound card calls can be either passed to the IHP device 106 or passed off to the PC sound card to be handled normally.

Switch condition is governed by the client software 110. When the switch is off (the normal condition) the NTMMM.DLL routes all sound signals to the PC sound card and all sound card is used instead of the IHP device 106 and telephone handset 106. If the switch is on as instructed by the client software 110, then the router 118 intercepts the group of sound signals that the VoIP engine 113 uses, and the NTMMM.DLL routes the resulting sound stream to/from the IHP device 106 via port 116.

The other key function of the router 118 is to extract state and command information from the serial data stream to/from the IHP device 106 and pass this to the IHP client software 110. The VoIP engine 113 only handles sound data from the DLL. The command data is routed to the client software 110, which then uses it to control the VoIP engine 113.

As an alternative to the foregoing embodiment, the router 118 may be eliminated completely by using a VoIP engine 113 that is specifically designed to interface with the IHP device 106 rather than a sound card. Still another alternative is to intercept the sound stream at the device driver level.

PC Software - IP Manager

The IP manager 112 performs conventional functions as found in other PCs connected to the Internet, and may comprise TCP/IP stack software, for example. Namely, the IP manager 112 breaks data to be transmitted into a fixed number of bytes. This packet of data is then inserted into a larger packet which includes the routing information for the data and additional information so that any errors in the transmission of the data can be detected. An example of a suitable protocol is TCP/IP. One example of the IP

manager 112 is the Dial Up Networking software included in the MICROSOFT WINDOWS operating systems.

Internet Home Phone (IHP) Device 106

IHP Device Hardware - Introduction

As mentioned above, the IHP device 106 performs a number of functions related to managing normal voice calls, Internet sessions, and concurrent voice/Internet connections. In the case of incoming or outgoing voice telephone calls (whether Internet or voice), the IHP device 106 provides a conduit between the telephone handset 108 and the remote telephone facilities 182.

When an Internet connection has not been previously established, the local PSTN 149a provides dial tone, busy signals, and ring signals in the earpiece for outgoing calls, ringing of the handset 108 for incoming calls, and the like. When the PC 102 has already started an Internet connection, however, IHP device 106 performs additional functions to simulate normal operation of the telephone handset 108 to conduct a VoIP call. For example, the IHP device 106 causes the telephone handset 108 to ring (for incoming calls), generates feedback signals for the IHP customer (such as ringing, busy signals, dial tone, etc.), receives and decodes the caller's dial pad entries (for outgoing calls), converts between analog signals at the telephone handset 108 and digital signals at the PC 102, and reroutes signals from the handset 108 to the PC 102 instead of directly to the remote telephone facilities 182. The IHP device 106 also selectively routes VoIP data from the PC 102 to the phone line 145 via the modem 104.

IHP Device Hardware - Details

The IHP device 106 includes a port 124, connectors 128 and 144, a modem 104 (optional), switches 140/142, an off-hook detection module 132, a tone decode/generation module 134, an analog-to-digital (ADC) and digital-to-analog (DAC) conversion module 136, ring generation module 138, and a processor 130.

As illustrated, the connectors 128/144 may comprise RJ-11 connectors suitable for connecting to telephone cords. Telephone cords couple the connector 128 to the handset 108, and also couple the connector 144 to a wall jack or other outlet attached to the remote telephone facilities 182. The connector 144 is selectively coupled to the connector 128 or modem 104 by switches 140, 142 as follows. The modem 104 is selectively coupled to the connector 144 when the switch 140 is closed, thereby coupling the modem 104 to the remote telephone facilities 182; the switch 140 may comprise a normally closed switch to provide failsafe features. The connector 128 is selectively coupled to the connector 144 when the switch 142 is closed, thereby connecting the handset 108 to the remote telephone facilities 182; the switch 142 is also normally closed. Upon power loss, both switches 140, 142 close, to institute a failsafe condition. The switches 140, 142 may be embodied by electrical relays, electromagnetic switches, or other suitable hardware.

The switches 140, 142 are coupled to the processor 130. The condition of the switches 140, 142 is set by the processor 130 by respective electrical inputs 140a, 142a to the switches. In one example, when power is applied to the IHP device 106, the processor 130 automatically configures the switch 140 into a default open position, and the switch 142 into a default closed position in order to anticipate normal use of the telephone handset 108 to place a standard voice telephone call by directly connecting to the facilities

182. As long as the handset 108 is not already engaged in a normal voice telephone conversation, the processor 130 opens the switch 142 and closes the switch 140 whenever the modem 144 goes "off-hook" to initiate an Internet connection. This disconnects the telephone handset 108 from the remote telephone facilities 182 forcing voice calls to be made by routing through the PC 102 according to VoIP technology. As discussed in greater detail below, a call diversion command may be implemented prior to the switch 140 being closed in order to enable the receipt of incoming VoIP calls to the handset 108.

The port 124 is coupled to the PC 102 by the cable 117, as mentioned above. The port 124 communicates with the PC 102 in order to relay digitally encoded voice and phone signalling data between the telephone handset 108 and the PC 102 via the IHP device 106. In the present example, the port 124 comprises an RS232 or USB type port, although other communications ports may be employed as discussed above in the context of the ports 114, 116.

The port 124 may advantageously include hardware to prevent dangerous voltages from being applied to the telephone line. For example, the port 124 may utilize a capacitively isolated supply to power line drivers and opto-isolators to pass the serial signal from the processor to the line drivers. The line drivers and optical isolators (not shown) are located between the port 124 and the processor 130.

Under current technology, serial port communication speed is limited to 115,200 bits/second by the WINDOW S operating system, allowing only one standard 64 Kbits/s data stream to be sent/received to/from the PC 102 by the port 124. Using USB or undertaking additional compression of the voice signal on the IHP device 106 may be performed to remove this limitation, and additionally support two or more sound streams.

The processor 130 manages operations of the IHP device 106, and may be embodied by a digital signal processor, or a variety of other hardware devices discussed in greater detail below. The processor 130 is responsible for various duties, including issuing the call diversion commands to the PSTN 146a (in one embodiment), detecting the on-hook/off-hook status of the modem 104 and handset 108, collecting decoded DTMF tones, and exchanging digitized voice data with the PC 102. In one specific embodiment, the processor 130 includes 128Kb of internal re-programmable flash memory (not shown) containing program code. If desired, this code may be upgraded in the field via download from the PC 102 via the serial communications cable 117, where the upgrade code is delivered to the PC 102 via the Internet, diskette, etc. In the illustrated embodiment, the processor 130 also includes 32Kb of SRAM (not shown) for program and data storage. An exemplary operating speed for the processor 130 is forty million instructions per second, i.e., 40 MIPS.

The off-hook detection module 132, connected to the processor 130, serves to notify the processor 130 whether (a) the telephone line 145 is already engaged by the modem 104, and also whether (b) the handset 108 is "off-hook" for example by the IHP customer lifting the handset 108 from its cradle. The processor 130 responds by commencing a call initialization sequence detailed in FIGURE 6A (described below). As one example, the module 132 may be coupled to the connector 128, where the module 132 operates by detecting changes in the electrical current in the telephone cables connected to circuitry of the handset 108, which provides signals indicative of whether the handset 108 is off-hook or on-hook. The module 132 may be implemented by known circuitry designs using

a combination of resistors, capacitors, inductors and transformers manufactured as a printed circuit board.

The tone decode/generation module 134, connected to the processor 130, provides audible feedback to the handset user via the handset ear piece. Such audible feedback, for example, includes dial tones, busy signals, ringing signals, and other signals that the module 134 generates to simulate normal use feedback that the handset 108 is operating normally. As one example, the module 134 may be implemented by circuitry such as a MyTel brand MT8870 circuit.

The analog-to-digital (ADC) and digital-to-analog (DAC) conversion module 136, connected to the processor 130, serves to convert analog voice speech (from the handset 108) into digital signals capable of processing by the processor 130 and IHP client software 110 for transmission as a VoIP call. It also converts the return digital signals from the PC 102 into analog signals suitable for transmission to the handset 108 as audible voice speech. As one example, the module 132 may be implemented by circuitry such as a Texas Instruments brand TLC 320AC50 circuit.

The ring generation module 138, connected to the processor 130, provides the required voltage signal pattern to cause the phone to ring in its normal manner to indicate an inbound call. As one example, the module 138 may be implemented by circuitry such as a transistor bridge driver circuitry, controlled by the processor 130.

In addition to the foregoing hardware components, the IHP device 106 also includes certain power circuitry (not shown). For example, power for the device 106 may be supplied by a an alternating current plug pack of nine volts, twelve volts, or another suitable low voltage. Such a pack is employed to generate the multiple voltage levels needed

within the device (e.g., +5, -5, +24, +90 Volts), as well as the isolated supply required to power the serial interface (e.g., +5, -5 Volts). This is done using voltage doubling techniques, the same technique also providing the isolated supply with the link capacitors providing the voltage isolation. If desired, the IHP device 106 may include heartbeat light-emitting diodes to indicate all power supplies are valid and that the processor and software are running correctly.

IHP Device Software

In one embodiment, the processor 130 may be programmed to provide a state machine that moves from state to state based on input stimuli. This stimuli comes from the off-hook detection module 132 (line hook detection signals), the tone decode/generation module 134 (decoded DTMF tones from the handset 108), and the PC 102 (control signals). In this manner, the processor 130 provides one state machine that operates in cooperation with another state machine implemented by software of the PC 102. As each state is entered and exited, outputs are set and cleared. This is done in such a way as to achieve the operating conditions explained below.

To enhance sound quality, the processor 130 may also perform added signal processing or such added processing may be performed by dedicated circuitry. For example, the processor 130 may compress/expand data signals to exchange data more quickly with the PC 102. As an example, the processor 130 may compand sound streams prior to transmitting them to the PC 102, and decompand sound streams received from the PC 102. For example, this companding may involve logarithmically compressing thirteen bits into eight bits to maintain dynamic range, providing a resulting sound stream of eight

KHz, eight bits per sample. The processor 130 may additionally be programmed to perform other signal processing, such as echo cancellation.

Alternatively, the processor 130 may compress the resultant sound stream using a Global System for Mobile telecommunications (GSM) voice compression algorithm or other suitable compression algorithm. Some exemplary compression options include G711 (a-law & mu-law), GSM FR, GSM EFR (such as G729a or G723.1), and the like. This compressed voice signal is then forwarded to the PC's IHP client software 110 for encapsulation in the Internet compatible packets for transmission via the IP manager 112.

Modification For Multiple Simultaneous Voice Conversations

For ease of explanation, the IHP device 106 is illustrated as supporting a single voice conversation. In an alternative embodiment, the IHP device 106 may be modified to support multiple bidirectional sound streams. To achieve this additional capability the IHP client software 110 and sound signal router 118 are modified to support multiple active implementations of the IHP client software 110 and the sound signal router 116. Additionally, in this embodiment the IHP device 106 is modified to include an additional connector (not shown) to support an additional telephone handset (not shown) and duplicates of the switch 142 and modules 132, 143, 136, 138. The processor 130 of this embodiment includes added programming to support the additional voice channel. In this enhanced configuration, a mini PBX capability may also be offered, enabling calls to be made between the handsets connected to the IHP device 106.

Exemplary Digital Data Processing Apparatus

As mentioned above, the IHP device 106 performs certain software functions, and may be implemented by a digital data processing apparatus configured to execute machine-readable code. FIGURE 2 shows a more specific example, in the form of the digital data processing apparatus 200. The apparatus 200 includes a processor 202, such as a microprocessor or other processing machine, coupled to a storage 204. In the present example, the storage 204 includes a fast-access storage 206, as well as nonvolatile storage 208. The fast-access storage 206 may comprise random -access memory ("RAM"), and may be used to store the programming instructions executed by the processor 202. The nonvolatile storage 208 may comprise, for example, one or more magnetic data storage disks such as a "hard drive", a tape drive, or any other suitable storage device. The apparatus 200 also includes an input/output 210, such as a line, bus, cable, electromagnetic link, or other means for the processor 202 to exchange data with other hardware external to the apparatus 200.

Despite the specific foregoing description, ordinarily skilled artisans (having the benefit of this disclosure) will recognize that the apparatuses discussed above may be implemented in machines of different construction, without departing from the scope of the invention. As a specific example, one of the components 206, 208 may be eliminated; furthermore, the storage 204 may be provided on-board the processor 202, or even provided externally to the apparatus 200.

Logic Circuitry

In contrast to the digital data storage apparatus discussed previously, a different embodiment of the invention uses logic circuitry instead of computer-executed instructions.

Depending upon the particular requirements of the application in the areas of speed, expense, tooling costs, and the like, this logic may be implemented by constructing one or more application-specific integrated circuits ("ASICs") having thousands of tiny integrated transistors. Such an ASIC may be implemented with CMOS, TTL, VLSI, or another suitable construction. Other alternatives include a digital signal processing chip ("DSP"), discrete circuitry (such as resistors, capacitors, diodes, inductors, and transistors), field programmable gate array ("FPGA"), programmable logic array ("PLA"), and the like.

OPERATION

In addition to the various hardware embodiments described above, a different aspect of the invention concerns a method for enabling the user of a standard telephone to make and receive phone calls over a telephone line that is already occupied by a connection between the user's PC and the Internet. The operation of the standard telephone handset is unchanged, yet the user of the computer is still able to browse the world wide web, upload or download files, and the like without affecting the quality of the voice telephone call.

Signal-Bearing Media

In the context of FIGURES 1A-1B, such a method may be implemented, for example, by operating the IHP device 106 and PC 102 to execute respective sequences of machine-readable instructions. These instructions may reside in various types of signal-bearing media. In this respect, one aspect of the present invention concerns one or more programmed products, each comprising a signal-bearing media tangibly embodying a

program of machine-readable instructions executable by a digital data processor to perform a method to enable the user of a standard telephone to make and receive voice phone calls over a telephone line that is already occupied by an Internet connection of the user's PC.

This signal-bearing media may comprise, for example, RAM (not shown) contained within the IHP device 106 or PC 102, as represented by the fast-access storage 206. Alternatively, the instructions may be contained in another signal-bearing media, such as a magnetic data storage diskette 300 (FIGURE 3). Whether contained in the storage 206, diskette 300, or elsewhere, the instructions may be stored on a variety of machine-readable data storage media, such as direct access storage (e.g., a conventional "hard drive", redundant array of inexpensive disks ("RAID"), or another direct access storage device ("DASD")), magnetic tape, electronic read-only memory (e.g., ROM, EPROM, or EEPROM), optical storage (e.g., CD-ROM, WORM, DVD, digital optical tape), paper "punch" cards, or other suitable signal-bearing media including transmission media such as digital and analog and communication links and wireless. In an illustrative embodiment of the invention, the machine-readable instructions may comprise software object code, compiled from a language such as "C" and Assembler, using the Texas Instruments development tools.

Logic Circuitry

In contrast to the signal-bearing medium discussed above, the method aspect of the invention may be implemented using logic circuitry, without using a processor to execute instructions. In this embodiment, the logic circuitry is implemented in the IHP device 106

and/or PC 102, and is configured to perform operations to implement the method of the invention. The logic circuitry may be implemented using many different types of circuitry, as discussed above.

Overview- Operating Modes

As an overview, the following operating modes describe the operation of the different components of the system 100.

No Internet Connection, IHP Customer Places a Voice Telephone Call

In this mode, when the caller lifts the telephone handset 108, the IHP device 106 automatically connects the handset 108 to the telephone line directly, and the caller places a normal voice telephone call.

No Internet Connection, IHP Customer Receives a Voice Telephone Call

In this mode, when the telephone 108 rings, the IHP device 106 passes this ring signal straight through to the telephone handset. The handset 108 rings normally, and when the customer answers the call it proceeds as normal, albeit via the IHP device 106.

With Existing Internet Connection, IHP Customer Places a Voice Telephone Call

When the customer operates the PC 102 to connect to the Internet, the PC 102 directs the modem 104 to dial the ISP 147a, and the processor 130 configures the switch 140 to couple the modem 104 to the remote telephone facilities 182 (including the voice gateway 146a) via the phone line 145. The processor 130 also configures the switch 142

to disconnect the handset 108 from the line 145. This prevents the customer from making a normal call using the handset 108.

To place a voice call, the customer lifts the handset 108 as normal. The off hook detection module 132 senses this, and alerts the processor 130. If the processor 130 verifies that the modem 104 is engaged in an Internet session (as in the current example), the processor 130 uses the tone decode/generation module 134 to send a simulated dial tone to the handset 108. Otherwise, if the modem 104 is engaged in other business than an Internet session (e.g., sending a fax, connecting to a dedicated non-Internet service, etc.), the processor 130 directs the module 134 to simulate a busy tone at the handset 108.

Upon hearing the simulated dial tone, the customer operates the handset dial pad to enter the number to be dialed. The tone decode/generation module 134 decodes this information, and the processor 130 passes the decoded, dialed telephone number to the IHP client software 110 via the sound signal router 118. The IHP client software 110 verifies the number dialed against a table of allowable dialed sequences that is accessible to (or even incorporated into) the IHP client software 110. The table of allowable dialed sequences is variable depending upon the numbering code selected by the local carrier or local telecommunications regulatory authority. Once a valid number is detected, the IHP client software 110 directs the VoIP engine 113 to (1) establish a session with the voice gateway 146a to create a logical connection between the PC 102 and the voice gateway 146a, and then (2) forward the dialed number to the voice gateway 146a.

The voice gateway 146a establishes a least cost route to the called number by reference to internal databases of the voice gateway 146a. This route may include a direct connection to the PSTN 149b or utilization of the infrastructure 155 to the terminating

gateway 146c or 146b. In this example, the "originating" voice gateway is 146a, and the "terminating" gateway is the voice gateway via which the call exits the voice gateway service providers infrastructure, such as gateway 146c or 146b.

The voice gateway 146a then routes the call to the required destination, which may occur using known principles of telephony. If connection is possible, the called party's PSTN (terminating PSTN) causes the called phone to ring. If the connection is not possible, for example if the called phone is off-hook or disconnected, then the called party's PSTN returns a message advising that the call has failed. In the case of the voice gateway 146b, the gateway 146b returns a busy or disconnect message to the calling party's voice gateway 146a, which proceeds to the IHP client software 110, which passes a representative signal to the calling party's IHP device 106, which in turn generates the appropriate tone to indicate the status of the line.

If the call is routed to an IHP-connected client, the terminating voice gateway (such as 146b) initiates a connection to the called party's VoIP engine (not shown), which sends an inbound call notification request to the called party's IHP device. At the called party's IHP device, the processor commands the ring tone generator to make the phone handset ring. If any module in the path from the terminating voice gateway 146b to the handset is unable to complete its task, the voice gateway sends an appropriate signal back to the originating voice gateway 146a. This signal is transmitted to the originating IHP 106 via the various intermediate modules that generates the appropriate tone to the telephone handset 108.

Once a complete route is established from the calling party's IHP device 106 to the called party, whether it is terminating via the PSTN or via an ISP, the calling party's IHP device 106 directs the module 138 to generate a ring tone on the calling party's handset

108. If the called party answers, the IHP client software 110, VoIP engine 113, voice gateways in the call path, and the PSTN (if applicable) cooperatively establish bi-directional communications between the parties. Then, the IHP device 106 and PC 102 convert voice speech from the calling party's handset 108 into IP packets. The PC 102 and IHP device 106 convert IP packets received from the voice gateway 146 into a format compatible with transmission to a normal telephone handset. If either party hangs up, the voice gateway that manages the connection with the disconnecting party detects the disconnection and generates a busy tone to the party left on the line. If the remaining party is the call originator, that party's IHP device generates a call-disconnected tone to the associated handset.

With Internet Connection, IHP Customer Receives a Voice Telephone Call

When the PC 102 is already connected to the Internet, incoming voice telephone calls are processed in the following way. Before initialization of the Internet connection, the IHP device 106 diverts the incoming voice calls to the voice gateway 146a associated with the IHP customer. Namely, the IHP device 106 or PC 102 directs the modem to transmit a call diversion command to the PSTN 149a, or this may be done manually by the IHP customer. The call diversion command instructs the PSTN 149a to divert all incoming calls to a specific telephone number, namely the number of the voice gateway 146a. An example of a suitable service supplied by a local exchange carrier is the call forwarding feature offered by Pacific Bell and other RBOC's. The call forwarding feature is initiated by dialing DTMF digits such as 72* followed by the voice gateway's phone number.

Later, when the PSTN 149a receives a call route request to send an incoming call to the IHP customer's telephone number, the PSTN 149a automatically diverts the call to the IHP customer's voice gateway 146a. As an example, the call forwarding (call diversion) number may be unique for each IHP customer, to enable the voice gateway 146a to easily determine which IHP customer is being called. As an alternative, the PSTN delivering the call to the voice gateway 146a may supply the phone number of the IHP customer being called to the voice gateway 146a, for example, by utilizing the internationally recognized common channel signaling system number 7 (CCSS7) signaling protocol. The voice gateway 146a processes the CCSS7 data by extracting the number that was called by the party that has had their call diverted to the voice gateway. Having identified the IHP customer's telephone number by whatever means, the voice gateway then uses a database of IHP customer's phone numbers versus IP addresses and other user details and routes the call to the IHP customer's PC 102. The voice gateway then uses its customer database to locate the customer that is connected to its ISP 149a using this phone number. The voice gateway then is able to locate the IP address of this customer and hence set up a route for the inbound call.

After call diversion, the voice gateway 146 undertakes the following functions. First, the voice gateway 146a converts the incoming voice call from a format compatible with the standard switched telephone network (e.g., ANSI, ETSI, etc.) to an Internet-compatible signal, namely an IP signal. Second, the voice gateway 146a routes the incoming voice call to the PC 102 over the PC's existing Internet connection from which the incoming call was originally diverted.

When the call route has been established to the IHP customer's PC 102, the IHP client software 110 communicates with the IHP device 106 over the cable 117 to indicate that there is an incoming call. In response, the processor 130 directs the ring generation module 138 to generate a normal ring signal at the handset 108. When answered, the two parties will be connected through the path that includes the handset 108, connector 128, conversion module 136, processor 130, port 124, cable 117, port 116, sound signal router 118, IHP client software 110, IP manager 112, port 114, modem 104, connector 144, line 145, local PSTN 149a, ISP 147a, voice gateway 146a and VoIP provider IP infrastructure 155. The voice gateway interconnects with the PSTN used by the calling party's PSTN to deliver the call.

So connected, the parties can converse as in a normal telephone call. When either side hangs up, the calling party's PSTN or the voice gateway 146 detects the termination by the party connected to its network, and then signals the other and terminates the call route. If the call has been terminated by the calling party, the voice gateway 146a signals the IHP client software 110 that in turn signals the IHP device 106. The IHP device 106 generates a call disconnected signal to the handset 108.

Failsafe Operation

When power to the IHP device 106 is disconnected, it is important for safety reasons that the telephone 108 still work in a reliable manner. When the power is lost, the telephone handset 108 and modem 104 are both connected to the telephone line 145 because of the normally closed configuration of the switches 140, 142. This way either of the handset 108 or modem 104 can make or answer a call.

Overall Sequence of Operation

FIGURE 4 shows a sequence 400 to illustrate the overall operation of the present invention. Broadly, the sequence 400 provides a method for interfacing a voice telephone with a telephone line regardless of whether that line is already occupied by a computer-to-Internet connection. For ease of explanation, but without any intended limitation, the example of FIGURE 4 is described in the context of the hardware environment 100 of FIGURES 1A-1B, as described above. The steps are initiated in step 402, when a customer, technician, or other operator interconnects the IHP device 106, handset 108, PC 102, and telephone line 145 as shown in FIGURE 1A. At this time or previously, the IHP user must become a customer of the voice gateway service provider associated with the voice gateway 146a. At this time (or previously), the operator may also install the software components 110, 113, 118, 112.

Then, the system 100 stands ready to conduct Internet data, VoIP telephone signals, or normal voice conversations over the telephone line 145. Step 404 senses whether an Internet connection or voice call has been initiated. For instance, the customer may activate the handset 108 to place or answer a call, since the normally-closed switch 142 connects the handset 108 to the household telephone line 145, allowing normal voice calls to begin (step 406). Conversely, the customer may lift the handset 108 to answer a normal voice call. Normal incoming and outgoing voice calls are connected using normal PSTN signalling, and telephone ringing (incoming calls) and ringing/busy signals (outgoing calls) in the handset earpiece are generated normally (step 406). As of step 406, an Internet connection cannot be initiated, since the Internet connection must be formed first to permit concurrent Internet and VoIP data exchange.

On the other hand, if the PC 102 in step 404 senses that the IHP customer has entered instructions to begin an Internet connection (such as by using a mouse or entering keyboard commands), then step 408 establishes an Internet data connection. If the IHP customer wishes to receive calls while connected to the Internet, the IHP customer must instruct the PC 102 as part of the Internet connection sequence to send a call diversion command to the PSTN 149a, or establish the call diversion manually prior to Internet connection.

In step 408, the PC 102 operates the port 114 and modem 104 to establish an Internet data connection through the PSTN 149a to the ISP 147a. At this time a session can be automatically started with the voice gateway 146a which authenticates the IHP customer as an agreed subscriber to the services of the voice gateway supplier. Following completion of the log in to the ISP 147a, the PC 102 automatically launches the IHP client software 110. The IHP client software 110 authenticates the IHP customer as a valid VoIP customer of the voice gateway supplier. This may be done, for example, by the IHP client software 110 obtaining a user name and password from the IHP customer, and then directing the VoIP engine 113 to transmit this data to the voice gateway 146a via the ISP 147a. The voice gateway 146a attempts to validate the user against its database, and if successful the voice gateway service provider server sends an authentication request accepted back to the VoIP engine 113. The VoIP engine 113 transmits the authentication acceptance to the IHP client software 110, which then enters its call ready state. In an alternative embodiment, the ISP 147a initiates the log on process upon detecting the customer's log on to the Internet.

After step 408, step 409 establishes a VoIP session. Namely, the VoIP engine 113 establishes a client-server session to the voice gateway 146a under the management of the client software 110. This operation may be implemented using conventional techniques, such as the known process to establish a session with an Ericsson IPT voice gateway. Further details of this operation will be apparent to an ordinarily skilled artisan, having the benefit of this disclosure.

After step 409, step 410 senses whether a voice call has been initiated. If not, step 412 asks whether the existing Internet connection has been terminated. This may occur by action of the PC user, dropped call, etc. When the PC 102 finds that the connection has been terminated, step 412 returns to step 404, discussed above. Otherwise, step 412 returns to step 410.

In contrast, step 410 proceeds to step 414 if a voice call has been initiated. Step 410 may sense initiation of a voice call by various means, such as the IHP device 106 sensing customer activation of the handset 108 to place an outgoing call, or the PC 102 receiving VoIP data representing an incoming call from the voice gateway 146a.

In the case of an incoming call, step 414 involves the following operations. Initially the IHP customer's local PSTN 149a routes the incoming call back to the voice gateway 146a due to the call diversion placed in step 408, which specified the forwarding telephone number of the voice gateway. The voice gateway 146a converts the inbound call from its PSTN format to VoIP format, and forwards the converted data to the IHP customer's Internet address. The voice gateway 146a may obtain the IHP customer's identity in various ways such as (1) because the diversion is to a unique number that the voice gateway recognizes as being associated with a particular user of its service, or (2) because

the call-originating PSTN attached the called number with its call request. The calling party and the called party information are delivered from the local exchange carrier to the voice gateway service provider via the CCSS7 signaling protocol as described elsewhere herein. Using the called number (of the IHP customer), the voice gateway 146a locates the Internet address of the IHP customer on an internal voice gateway database and then routes the incoming call notification to the PC 102 via the ISP 147a.

If the IHP client log on process to the voice gateway authentication server has not been successfully completed, the voice gateway 146a returns a line busy via the CCSS7 signaling system to the calling party's PSTN.

On the other hand, if the PC 102 is already running the IHP client software 110, the VoIP engine 113 detects the incoming VoIP call from the voice gateway 146a, whereupon the client software 110 alerts the processor 130 of the IHP device 106 via the cable 117. In response, the processor 130 operates the ring generation module 138 to cause the handset 108 to ring. When the off-hook detection module 132 of the IHP device 106 senses that the customer has lifted the handset 108 to answer the incoming call, the IHP device 106 and PC 102 cooperatively convert sound from the handset 108 into VoIP data and transmit the data to the dialed number. Simultaneously, the IHP 106 and PC 102 convert digital data received via the gateway 146a into analog output signals, which are transmitted to the handset 108. More specifically, incoming VoIP data (i.e., the outside party's voice) is received along with the non-VoIP Internet data at the modem 104, extracted from the Internet data by the IP manager, the VoIP data re-formatted by the VoIP engine 113 to remove the Internet compatible encapsulation used for transmission of the voice information over the Internet, routed by the sound signal router 118 to the IHP device

106 via the cable 117, and converted into analog signals by the conversion module 136 for transmission to the handset 108. Outgoing VoIP data (i.e., the customer's voice) proceeds in the opposite path.

In contrast to the foregoing description, step 414 involves the following operations in the case of outgoing calls. Initially, the off-hook detection module 132 of the IHP device 106 detects when the customer activates the handset 108 to initiate an outgoing call. In response, the processor 130 directs the tone decode/generation module 134 to provide a simulated dial tone at the handset 108. Also, the tone decode/generation module 134 receives and decodes pushbutton signals created when the customer dials a telephone number, and hands the decoded signals to the processor 130. The IHP device 106 passes the decoded telephone number to the IHP client software 110, which initiates a call to the dialed number by forwarding a request for connection to the voice gateway 146a, which includes the identity of the dialed number. The voice gateway 146a establishes a path to the dialed number over a combination of its own infrastructure 155, non-VoIP infrastructure 180, and terminating PSTNs as appropriate. When the call is connected, the terminating voice gateway (e.g. 146b-146c) is notified via the CCSS7 signaling protocol from the terminating PSTN, whereupon the voice gateway service provider notifies the VoIP engine 113 and client software 110 via a combination of CCSS7 and proprietary signaling protocols; the IHP client software 110 in turn notifies the IHP device 106, which operates the tone decode/generation module 134 to simulate a ringing or busy sound in the telephone handset 108 earpiece, depending upon whether the dialed number is ringing or busy. If the dialed number is answered, the IHP device and VoIP engine 113 cooperatively convert sound from the handset 108 into VoIP data, the IP manager adds

the VoIP data to the existing Internet connection to transmit the data to the dialed number. Simultaneously, the IHP device 106 and PC 102 convert digital data received from the dialed number via the gateway 146a into analog output signals, which are transmitted to the handset 108.

After the VoIP connection is established in step 414, step 416 asks whether the VoIP or data connection has been terminated. Termination of the VoIP connection may be sensed, for example, by the voice gateway 146a (1) because it receives notification of termination by the other party via the CCSS7 signalling protocol, or (2) because the IHP device 106 detects the phone going back on hook through use of the module 132, and the IHP device 106 responsively notifies the IHP client software 110, whereupon the IHP client software 110 via the VoIP engine 113 notifies the voice gateway 146a. Termination of the data connection may be sensed, for example, (1) by the Internet connection software of the PC 102, which notifies the IHP client software 110, which in turn notifies the IHP device 106, which provides a busy signal to the phone handset 108 via the conversion module 136, or (2) by the voice gateway 146a, which notifies the PSTN 149a and any other facilities 180 involved in routing the call by forwarding the appropriate CCSS7 based message, whereupon the PSTN 149a and any other applicable facilities forward a call busy signal to the remote party.

If the VoIP connection has been terminated, step 416 ends the call at the IHP device 106, as discussed in greater detail below in FIGURES 9-10. Then, step 416 returns to step 410 to await initiation of a new VoIP connection. In contrast, if the data/Internet connection has been terminated, this also has the effect of terminating the VoIP call because the VoIP service requires continuous operation of the connection between the PC

102 and the ISP 147a. Accordingly, both data and VoIP calls end in step 422. At this point, the system 100 is not maintaining any VoIP or data connections. Accordingly, step 422 returns to step 404 to await a new voice or data call.

Detailed Operation

Having set forth a general overview of the invention, more detailed explanation is now made with references to remaining FIGURES 5-12. For ease of illustration, these drawings are discussed in the context of the illustrative system 100 of FIGURE 1A-1B, as described above.

Initial Log-On (FIGURE 5)

FIGURE 5 describes a sequence 500 whereby a customer employs the components of the system 100 to log-on to the Internet, as a more detailed example of step 408 (FIGURE 4). One important feature of the invention is that, prior to logging on to the Internet, the IHP device 106 or PC 102 initiates a call diversion command to the local PSTN 149a, which diverts any incoming calls to the voice gateway 146a along path 170; thus, these calls can be received by the customer over the customer's existing Internet connection in IP format where they are then processed by the PC 102 and IHP device 106 to present inbound calls to the phone handset 108.

Referring to FIGURE 5, the sequence 500 begins in step 502, where the IHP customer initiates a connection to the Internet. This is achieved by activating graphical user interface (GUI) of the IHP client software 110.

In step 504, the IHP client software 110 software determines whether, during the installation operations of step 402 (or any subsequent reconfiguration), the customer has previously revealed any preferences regarding "call diversion." Call diversion concerns the diversion of incoming voice calls to the IHP device 106 in order to receive incoming voice calls using the handset 108 during the Internet connection. In the illustrated embodiment, depending upon the customer's preferences, the client software 110 may (1) always institute call diversion before connecting to the Internet (step 508), (2) never utilize call diversion (skip to step 510), or (3) on a call-by-call basis, utilize a graphics display upon the PC 102 to prompt the user for his/her call diversion wishes (step 506). If the customer does not select call diversion, call diversion is not instituted, and the processor 500 advances to step 510 (discussed below) to begin the Internet connection. In this case, the IHP customer will not be able to receive voice calls on the handset 108 during his/her Internet connection. Incoming calls will encounter a busy signal.

On the other hand, if step 504 or 506 finds that the IHP customer has selected call diversion, call diversion is implemented in step 508. Call diversion, also known as "call forwarding," is a local PSTN option whereby PSTN customers can have incoming calls directed to another telephone number, either a fixed, preselected number or a number selected for that instance of call forwarding. Although the details of this command varies from country to country, and even varies among local PSTN services, many instances involve transmitting DTMF tones by making telephone keypad entries in response to a voice menu provided by the PSTN company. If the call diversion option is not available at the PSTN 149a, the inbound voice calls cannot be received by the IHP device 106.

In step 508, the IHP client software 110 transmits instructions to the processor 130 to carry out the call diversion command, as discussed below in greater detail. The call diversion command instructs the local PSTN 149a to divert incoming calls to a predetermined number, at the voice gateway 146a. The appropriate call-forwarding number depends upon whether or not the local exchange carrier (LEC) includes as part of the CCSS7 data that it forward to the voice gateway service provider the called number in addition to the calling number. The supply of this information by the LEC is dependent upon the legislative environment within which the LEC operates and the contractual relationship between the LEC and the voice gateway service provider. If the LEC supplies the called number, then the call forward number is the a general purpose voice gateway inbound number. If the called number is suppressed by the LEC the voice gateway service provider will supply a unique indial number to the IHP customer that is within its indial range. When the call forward occur it will be to this unique number. The voice gateway service provider on the basis of the number called is then able to correlated the number called from its customer database.

Call diversion, in one example, may be implemented by the following process. First, the client software 110, recognizing that call diversion is to be applied (step 508), advises the processor 130. In response, the processor 130 generates the appropriate call diversion codes (such as *72) compatible with the PSTN 149a. Then, the processor 130 (1) closes the switch 142 to connect the DAC module 136 to the line 145, (2) directs the tone decode and generation module 134 to convert the call diversion code into DTMF tones for transmission to the PSTN 149a, and (3) after successful completion of the call diversion sequence, opens the switch 142 and closes the switch 140. According to a

different example, call diversion may be carried out by the PC 102 commanding the modem 104 to transmit a DTMF call diversion command via the line 145.

After establishing call diversion (step 508), or after determining that call diversion is not desired (step 506), step 510 initiates the dial-up Internet connection. Namely, the client software 110 invokes Internet connection software of the PC 102 such as the MICROSOFT Dial Up Networking software component. This component transmits instructions to the modem 104 to dial the IHP customer's ISP 147a. In step 512, the ISP 147a authenticates the user as a valid customer of the ISP's Internet services. If authentication fails for any reason, the IHP initialization process is aborted.

Following successful authentication to the ISP 147a occurring in step 512, the routine 500 ends in step 518.

Outgoing Call Sequence (FIGURES 6A-6C)

FIGURES 6A-6C collectively describe a sequence 600 whereby a customer employs the components of the system 100 to place an outgoing voice call. The sequence 600 provides a more detailed example of tasks 410, 414 and 416 (FIGURE 4). Advantageously, the sequence 600 recognizes when the telephone line 145 is occupied by an existing Internet connection, and takes appropriate action to establish a VoIP call rather than a normal voice call.

Referring to FIGURES 6A-6C, the sequence 600 begins in step 602 where the IHP customer lifts the handset 108 from its cradle. The following actions depend upon whether power is applied to the IHP device 106 or not (step 604). If power is not applied to the IHP device 106, the switches 140, 142 will be in their fail safe position, i.e both closed (step

610). In this condition, if the PC 102 is not already conducting an Internet session, the handset 108 will operate as normal (step 612). If the PC 102 is already connected to the Internet, then the IHP customer user will not receive a dial tone, and s/he will be unable to make a voice call with the handset 108 in step 612.

On the other hand, if the IHP device 106 is powered up, then step 604 advances to step 606. In step 606, the off hook detection module 132 detects that the IHP customer has lifted the handset 108, and sends a representative message to the processor 130. When this occurs, the processor 130 determines whether the modem 104 is already occupying the line 145 (step 608). This is achieved by the off hook detection circuitry 132 that senses whether the phone line 145 is being used by the modem 104. If the modem 104 is not being utilized, then the IHP device 106 permits the handset 108 to operate normally to conduct a standard voice call (step 612).

If the modem 104 is occupying the line 145, however, then step 608 advances to step 614. At step 614, the processor 130 advises the IHP client software 110 of the handset 108 being lifted, and the software 110 proceeds to verify that a VoIP session has been established as described in detail in step 409 (FIGURE 4), and attempts to complete this process if not already done (step 616). Step 622 repeats the inquiry of step 614, and if the VoIP session has failed, the software 110 advises the processor 130, which causes the tone decode/generation module 134 to transmit a busy signal to the handset 108 earpiece (step 626), and the routine 600 ends (step 627).

On the other hand, if the IHP client software 110 and VoIP engine 113 succeed in connecting to the voice gateway 146a, step 622 (or step 614) advances to step 620. In step 620, the client software 110 advises the processor that the Internet connection is in

place, and the processor 130 directs the tone decode/generation module 134 to generate a dial tone at the handset 108, for the benefit of the IHP customer. In steps 628, 630, the processor 130 and tone decode/generation module 134 monitor the handset 108 to capture any DTMF digits as they are dialed by the IHP customer. If the IHP customer dials a digit, step 628 proceeds to step 632 (FIGURE 6B), as described below. If the IHP customer does not dial a digit with a predetermined time ($\frac{1}{2}$ second in this example), then step 630 advances to step 636 (FIGURE 6B), as described below.

In step 632 (FIGURE 6B), the tone decode/generation module 134 decodes the dialed digit (from DTMF tones), and the processor 130 then forwards the dialed digit to the IHP client software 110. Then, the IHP client software 110 determines whether the digits dialed thus far represent a valid telephone number (step 634), which is conducted by comparing the dialed digits to an internal database that is integral to the IHP client software to determine whether the digits represent a legitimate telephone number. If the dialed digits do not yet represent a valid telephone number, step 634 returns to step 630 (FIGURE 6A) to await the next digit.

When the IHP client software 110 finds that the dialed digits represent a valid telephone number, step 634 advances to step 636. Namely, at step 636 the IHP client software 110 directs the VoIP engine 113 as a client of the voice gateway 146a to transmit the dialed digits to the voice gateway 146a. The voice gateway 146a attempts to establish a call route to the dialed number in step 638. The route selected may utilize any of the telecommunications facilities available to the voice gateway service provider.

After step 638, the voice gateway 146a asks whether the connection between the VoIP engine 113 and dialed number has been established (step 640). If the voice gateway

146a cannot establish a route, the voice gateway 146a returns an appropriate signal (e.g., busy, invalid number, disconnected line, etc.) to the VoIP engine 113. In step 642, the IHP client software 110 detects the returned signal, forwards an appropriate message to the processor 130, and the processor 130 directs the tone decode/generator module 134 to provide an appropriate tone (e.g., busy or call failed sound) to the phone handset 108 (step 642). After step 642, the routine 600 ends (step 627).

On the other hand, if the route is established (step 640), the voice gateway 146a using the IP protocol and proprietary signaling protocols transmits a signal to the VoIP engine 113 stating that the connection has succeeded and providing an indication of the status of the called party (e.g., ringing, busy, disconnected, non-working number, etc.) to the IHP client software 110 via the VoIP engine 113 (step 644). Also in step 644, the client IHP software 110 sends an IHP proprietary command to the processor 130 that indicates the completed status of the phone connection. In response, the processor 130 commands the module 134 to provide a ringing sound to the handset 108.

If the called party does not answer within a specified time, the voice gateway 146a executes a timeout, ending the connection (step 645). The VoIP engine 113 detects the call failure, whereupon the software 110 notifies the processor 130, which causes the module 134 to issue a busy signal. In contrast, if the called party accepts the call by lifting his/her telephone handset or other similar action, the acceptance of the call is signaled back to the voice gateway 146a by equipment such as the infrastructure 180, PSTN 149a, etc., as appropriate. The VoIP engine 113 signals the IHP client software 110 that the call has been completed. In response, the IHP client software 110 performs step 648.

In step 648, the IHP client software 110 sends a message to the processor 130 that the call has been answered; in response, the processor 130 commands the module 134 to stop the ringing sound in the handset 108. After step 648, step 649 establishes a bidirectional signal path locally in the PC 102 and IHP device 106. In particular, the client software 110 establishes a bidirectional path between the VoIP engine 113 and the sound signal router 118 for digitized voice signals. Also, the client software 110 commands the processor 130 to begin utilizing the DAC/ADC module 136 to convert between the analog signals at the handset 108 and digitized signals at the port 124.

After step 649, steps 650-653 and steps 660-663 are performed in parallel. Steps 650-653 describe the processing and transmission of the called party's voice to the IHP customer. Steps 660-663 describe the processing and transmission of the IHP customer's voice to the called party.

Steps 650-653 are described as follows. In step 650, the called party commences conversation. In response, various components of the system 100 route signals representing the called party's voice to the IHP customer's line 145 and then the PC 102 and IHP device 106 cooperatively translate these signals into voice signals heard upon the handset 108 (step 651). The details of step 651 are described below by the sequence 800 (FIGURE 8). After step 651, step 652 determines whether the called party has hung up. When the called party hangs up, step 652 advances to step 653, which performs a call termination sequence as described in detail by the sequence 900 (FIGURE 9).

As mentioned above, steps 660-663 occur in parallel with steps 650-653. In step 660, the IHP customer commences talking using the handset 108. In response, various components of the invention convert the customer's voice into VoIP signals and route

these signals to the called party's phone (step 661). The details of step 661 are described below by the sequence 700 (FIGURE 7). Following step 661, step 662 determines whether the IHP customer has hung up. Step 662 involves the off-hook detection module 132 determining that the handset 108 has been placed on hook, the module 132 notifying the processor 130, and the processor 130 advising the IHP client software 110. In response, the IHP client software 110 performs step 663, which comprises a call termination sequence as described in detail by the sequence 1000 (FIGURE 10).

Voice to IP Conversion Process (FIGURE 7)

FIGURE 7 describes a sequence 700 whereby various components of the system 100 translate the customer's voice into an appropriate signal and relay this signal to the called party. More particularly, in step 702 the ADC module 136 converts analog speech signals from the handset microphone 108 into a digital bit stream. Optionally, the processor 130 may also compand the digitized voice stream in step 702. Further processing may also be performed, if desired, to compress the companded digitized signal, perform echo cancellation, etc. Next, in step 704, the processor 130 transmits the digitized voice signal to the PC 102 via the serial communications cable 117. Also in step 704, the sound signal router 118 directs the digitized voice stream to the VoIP engine 113. In one example, this is achieved by the router 118 appearing to the VoIP engine 113 as the sound card on the PC 102. The operation of any PC sound card is not affected by this modification and is capable of operating normally while the IHP process is running.

In step 706, the VoIP engine 113 receives the digital voice stream and converts it into an IP compatible data-gramme packets, for example using a GSM codec. To improve

sound quality, the client software 110 may direct the VoIP engine 113 to ensure that these packets have a maximum packet length of 400 bytes to ensure that the both voice and web browsing can occur without having a negative impact on the quality of the voice.

In step 708, The VoIP engine 113 forwards the processed VoIP packets to IP manager 112, which transmits them to the voice gateway 146a. In response, the voice gateway 146a transmits the VoIP signals to the called party (step 710) using appropriate facilities. Step 710 completes the sequence 700.

VoIP to IHP Voice Delivery Path Sequence (FIGURE 8)

FIGURE 8 describes a sequence 800 whereby various components of the system 100 route the called party's voice to the IHP customer's line 145 and thereafter translate the customer's voice into an analog voice signals at the handset 108. In step 802, appropriate components of the remote telephone facilities 182 route the called party's voice to the customer's line 145. In step 804, the IHP device 106 carries the incoming VoIP signal from the line 145 to the PC 102 via the modem 104.

The IP manager 112 receives these packetized IP voice data and proceeds to process these signals in step 806. More particularly, in step 806, the IP manager 112 extracts the VoIP packets from other data on the Internet connection, and gives the packets to the VoIP engine 113. The VoIP engine 113 coverts these IP signals into digital, non-IP signals. Some examples of these resultant digital non-IP signals include a compounded thirteen bit 64k data stream, G711 GSM, or others depending upon which compression processes (if any) are utilized by the processor 130. Also in step 806, the

VoIP engine 113 transmits the digital, non-IP signals to a PC sound card, which are intercepted and re-routed by the sound signal router 118 to the IHP 106 via the port 116.

In step 808, the IHP device 106 converts the digital signals from the VoIP engine 113 into analog voice signals for transmission to the handset 108. Specifically, the processor 130 decompresses (optionally) the digitized voice stream, and routes this signal to the DAC module 136, which recreates analog speech. The DAC module 136 also creates the correct voltage and current levels to ensure that the analog signal is fully compatible with the phone handset 106. In step 810, the DAC module 136 outputs the analog voice signal to the handset 108, completing the sequence 800.

Sequence For Call Termination By Remote Party (FIGURE 9)

FIGURE 9 describes a sequence 900 whereby a VoIP call is terminated by the remote party, namely, the party that has been speaking with the IHP customer. Referring to FIGURE 9, the sequence 900 begins in step 902 where the ingress voice gateway (such as 146b) detects that the remote party has terminated the call, and signals the IHP customer's voice gateway 146a. In step 904, the IHP customer's voice gateway 146a transmits a representative message to the VoIP engine 113, which is forwarded to the supervising IHP client software 110. In response, the client software 110 terminates the call on the PC 102 by signalling to the VoIP engine 113 to terminate the session with the voice gateway 146a (step 906).

Also in step 906, the client software 110 instructs the processor 130 to also terminate the call. In particular, the client software 110 transmits a representative message to the processor 106, which responds by providing audible feedback of the terminated call

to the IHP customer. Namely, the processor 130 sends an instruction to the tone decode and generation module 134 to create a simulated busy tone at the handset 108. Step 908 completes the routine 900.

Sequence For Call Termination By IHP Customer (FIGURE 10)

FIGURE 10 describes a sequence 1000 whereby a VoIP call is terminated by the IHP customer. Referring to FIGURE 10, the sequence 1000 begins in step 1002 where, responsive to the IHP customer hanging up the handset 108, the off-hook detection module 132 sends an on-hook signal to the processor 130. In response, the processor 130 stops the processes of supervising analog-digital conversion, compression, relaying voice signals to/from the PC 102, etc. The processor 130 also transmits a call termination message to the IHP client software 110.

In response, the PC performs step 1004. Namely, the IHP client software 110 sends a terminate call command to the VoIP engine 113, whereupon the VoIP engine 113 notifies the voice gateway 146a of call termination. In step 1006, the PC 102 dismantles the path to the voice gateway 146a, and commands the IP manager 112 to increase the size of the IP packets from 400 bytes to the packet size required by the ISP 147a for the optimum transmission of Internet traffic over its network, such as 1,500 bytes. In response, the voice gateway 146a and other remote telephone facilities 182 end the connection between the IHP customer and the remote party (step 1008). This ends the routine 1000.

Incoming Call Sequence (FIGURE 11)

FIGURE 11 describes a sequence 1100 whereby a remote party places a call to the IHP customer, and the system 100 processes and routes the call to the IHP customer despite the existence of a previous Internet connection on the line 145. Referring to FIGURE 11, the sequence 1100 begins in step 1102 where the IHP customer's regional telephone company, such as the PSTN 149a, receives a request to route an inbound call to the IHP customer. The PSTN 149a determines whether the line 145 is in use (step 1104). If not, the call is completed normally (step 1106).

If the line 145 is occupied, however, step 1104 proceeds to step 1108, which asks whether a call diversion 170 is in place. If not, the PSTN 149a returns a busy signal to the calling party (step 1110). If the IHP customer has implemented the call diversion 170, step 1108 progresses to step 1112. In step 1112, the PSTN 149a requests that the voice gateway 146a accept the call. The voice gateway 146a accepts the call and, in response, proceeds to identify the IHP customer. In one embodiment, the voice gateway 146a determines the called number using the signalling protocol (such as CCSS7) used by the PSTN 149a, and then cross-references the called number against an internal database of IHP customer's IP addresses. In another embodiment, the voice gateway 146a takes the voice gateway 146a telephone number that the call has been diverted to, and cross-references this telephone number with an internal database to yield the IHP customer's IP address. In this second embodiment, each IHP customer has a different call forwarding number at the voice gateway 146a.

After step 1112, the voice gateway 146a sends a call request to the VoIP engine 113 via the ISP 147a, to which the IHP customer is already connected (step 1114). In response, the VoIP engine 113 sends notification of the incoming VoIP data to the IHP

client software 110, whereupon the client software 110 transmits notification to the IHP processor 130 (step 1116).

In response to step 1116, the processor 130 causes the ring generation module 138 to ring the handset 108 to indicate an incoming call (step 1118). When the IHP customer activates the handset 108 (step 1120), this is detected by the off-hook detector 132, which sends a representative command to the processor 130 (step 1120). In response, the processor 130 causes the ring generation module 138 to stop ringing the handset. The processor also sends a call accepted message to the IHP client software 110, completing step 1120.

After step 1120, the bidirectional VoIP call is conducted by performing steps 649 and following, as shown in FIGURE 6C.

OTHER EMBODIMENTS

While the foregoing disclosure shows a number of illustrative embodiments of the invention, it will be apparent to those skilled in the art that various changes and modifications can be made herein without departing from the scope of the invention as defined by the appended claims. Furthermore, although elements of the invention may be described or claimed in the singular, the plural is contemplated unless limitation to the singular is explicitly stated. Additionally, ordinarily skilled artisans will recognize that operational sequences must be set forth in some specific order for the purpose of explanation and claiming, but the present invention contemplates various changes beyond such specific order.

TABLE 1 – NTMMM.DLL

WINMMAPI BOOL WINAPI PlaySoundA(LPCSTR pszSound, HMODULE hmod, DWORD fdwSound)
 WINMMAPI MMRESULT WINAPI mixerClose(HMIXER hmx)
 WINMMAPI MMRESULT WINAPI mixerGetDevCapsA(UINT uMxId, LPMIXERCAPSA pmxcaps, UINT cbmxcaps)
 WINMMAPI MMRESULT WINAPI mixerGetID(HMIXEROBJ hmxobj, UINT FAR *puMxId, DWORD fdwId)
 WINMMAPI MMRESULT WINAPI mixerGetLineInfoA(HMIXEROBJ hmxobj, LPMIXERLINEA pmxl, DWORD fdwInfo)
 WINMMAPI UINT WINAPI mixerGetNumDevs(void)

 WINMMAPI MMRESULT WINAPI mixerGetControlDetailsA(HMIXEROBJ hmxobj, LPMIXERCONTROLDETAILS pmxcd, DWORD fdwDetails)
 WINMMAPI MMRESULT WINAPI mixerGetLineControlsA(HMIXEROBJ hmxobj, LPMIXERLINECONTROLSA pmxlc, DWORD fdwControls)
 WINMMAPI MMRESULT WINAPI mixerOpen(LPHMIXER phmx, UINT uMxId, DWORD dwCallback, DWORD dwInstance, DWORD fdwOpen)
 WINMMAPI MMRESULT WINAPI mixerSetControlDetails(HMIXEROBJ hmxobj, LPMIXERCONTROLDETAILS pmxcd, DWORD fdwDetails)

 WINMMAPI MMRESULT WINAPI timeKillEvent(UINT uTimerID)
 WINMMAPI MMRESULT WINAPI timeSetEvent(UINT uDelay, UINT uResolution, LPTIMECALLBACK fptc, DWORD dwUser, UINT fuEvent)
 WINMMAPI MMRESULT WINAPI waveInAddBuffer(HWAVEIN hwi, LPWAVEHDR pwh, UINT cbwh)
 WINMMAPI MMRESULT WINAPI waveInClose(HWAVEIN hwi)
 WINMMAPI MMRESULT WINAPI waveInGetDevCapsA(UINT uDeviceID, LPWAVEINCAPSA pwic, UINT cbwic)
 WINMMAPI UINT WINAPI waveInGetNumDevs(void)
 WINMMAPI MMRESULT WINAPI waveInOpen(LPHWAVEIN phwi, UINT uDeviceID, LPCWAVEFORMATEX pwfx, DWORD dwCallback, DWORD dwInstance, DWORD fdwOpen)
 WINMMAPI MMRESULT WINAPI waveInPrepareHeader(HWAVEIN hwi, LPWAVEHDR pwh, UINT cbwh)
 WINMMAPI MMRESULT WINAPI waveInReset(HWAVEIN hwi)
 WINMMAPI MMRESULT WINAPI waveInStart(HWAVEIN hwi)
 WINMMAPI MMRESULT WINAPI waveInUnprepareHeader(HWAVEIN hwi, LPWAVEHDR pwh, UINT cbwh)
 WINMMAPI MMRESULT WINAPI waveInStop(HWAVEIN hwi)
 WINMMAPI MMRESULT WINAPI waveInGetPosition(HWAVEIN hwi, LPMMTIME pmmt, UINT cbmmt)
 WINMMAPI MMRESULT WINAPI waveOutClose(HWAVEOUT hwo)
 WINMMAPI MMRESULT WINAPI waveOutGetDevCapsA(UINT uDeviceID, LPWAVEOUTCAPSA pwoc, UINT cbwoc)
 WINMMAPI UINT WINAPI waveOutGetNumDevs(void)
 WINMMAPI MMRESULT WINAPI waveOutOpen(LPHWAVEOUT phwo, UINT uDeviceID, LPCWAVEFORMATEX pwfx, DWORD dwCallback, DWORD dwInstance, DWORD fdwOpen)
 WINMMAPI MMRESULT WINAPI waveOutPause(HWAVEOUT hwo)
 WINMMAPI MMRESULT WINAPI waveOutPrepareHeader(HWAVEOUT hwo, LPWAVEHDR pwh, UINT cbwh)
 WINMMAPI MMRESULT WINAPI waveOutReset(HWAVEOUT hwo)
 WINMMAPI MMRESULT WINAPI waveOutRestart(HWAVEOUT hwo)
 WINMMAPI MMRESULT WINAPI waveOutUnprepareHeader(HWAVEOUT hwo, LPWAVEHDR pwh, UINT cbwh)
 WINMMAPI MMRESULT WINAPI waveOutWrite(HWAVEOUT hwo, LPWAVEHDR pwh, UINT cbwh)
 WINMMAPI MMRESULT WINAPI CloseDriver(HDRVR hDriver, LONG lParam1, LONG lParam2)

```

WINMMAPI HDRVR WINAPI OpenDriver(LPCWSTR szDriverName, LPCWSTR szSectionName, LONG
    IParam2)
WINMMAPI LRESULT WINAPI SendDriverMessage(HDRVR hDriver, UINT message, LONG IParam1,
    LONG IParam2)
WINMMAPI HMODULE WINAPI DrvGetModuleHandle(HDRVR hDriver)
WINMMAPI HMODULE WINAPI GetDriverModuleHandle(HDRVR hDriver)
WINMMAPI LRESULT WINAPI DefDriverProc(DWORD dwDriverIdentifier, HDRVR hdrv, UINT uMsg,
    LPARAM IParam1, LPARAM IParam2)
WINMMAPI LRESULT WINAPI DrvClose(HDRVR hdrv, LPARAM IParam1, LPARAM IParam2)
WINMMAPI HDRVR WINAPI DrvOpen(LPCSTR szDriverName, LPCSTR szSectionName, LPARAM
    IParam2)
WINMMAPI LRESULT WINAPI DrvSendMessage(HDRVR hdrv, UINT uMsg, LPARAM IParam1, LPARAM
    IParam2)
WINMMAPI LRESULT WINAPI DrvDefDriverProc(DWORD dwDriverIdentifier, HDRVR hdrv, UINT uMsg,
    LPARAM IParam1, LPARAM IParam2)
WINMMAPI UINT WINAPI mmsystemGetVersion(void)
WINMMAPI BOOL WINAPI sndPlaySoundA(LPCSTR pszSound, UINT fuSound)
WINMMAPI BOOL WINAPI sndPlaySoundW(LPCWSTR pszSound, UINT fuSound)
WINMMAPI BOOL WINAPI PlaySoundW(LPCWSTR pszSound, HMODULE hmod, DWORD fdwSound)
WINMMAPI BOOL WINAPI PlaySound(LPCSTR pszSound, HMODULE hmod, DWORD fdwSound)
WINMMAPI MMRESULT WINAPI waveOutGetDevCapsW(UINT uDeviceID, LPWAVEOUTCAPSW pwoc,
    UINT cbwoc)
WINMMAPI MMRESULT WINAPI waveOutGetVolume(HWAVEOUT hwo, LPDWORD pdwVolume)
WINMMAPI MMRESULT WINAPI waveOutSetVolume(HWAVEOUT hwo, DWORD dwVolume)
WINMMAPI MMRESULT WINAPI waveOutGetErrorTextA(MMRESULT mmmrError, LPSTR pszText, UINT
    cchText)
WINMMAPI MMRESULT WINAPI waveOutGetErrorTextW(MMRESULT mmmrError, LPWSTR pszText, UINT
    cchText)
WINMMAPI MMRESULT WINAPI waveOutBreakLoop(HWAVEOUT hwo)
WINMMAPI MMRESULT WINAPI waveOutGetPosition(HWAVEOUT hwo, LPMMTIME pmmmt, UINT cbmmt)
WINMMAPI MMRESULT WINAPI waveOutGetPitch(HWAVEOUT hwo, LPDWORD pdwPitch)
WINMMAPI MMRESULT WINAPI waveOutSetPitch(HWAVEOUT hwo, DWORD dwPitch)
WINMMAPI MMRESULT WINAPI waveOutGetPlaybackRate(HWAVEOUT hwo, LPDWORD pdwRate)
WINMMAPI MMRESULT WINAPI waveOutSetPlaybackRate(HWAVEOUT hwo, DWORD dwRate)
WINMMAPI MMRESULT WINAPI waveOutGetID(HWAVEOUT hwo, LPUINT puDeviceID)
WINMMAPI MMRESULT WINAPI waveOutMessage(HWAVEOUT hwo, UINT uMsg, DWORD dw1,
    DWORD dw2)
WINMMAPI MMRESULT WINAPI waveInGetDevCapsW(UINT uDeviceID, LPWAVEINCAPSW pwic, UINT
    cbwic)
WINMMAPI MMRESULT WINAPI waveInGetErrorTextA(MMRESULT mmmrError, LPSTR pszText, UINT
    cchText)
WINMMAPI MMRESULT WINAPI waveInGetErrorTextW(MMRESULT mmmrError, LPWSTR pszText, UINT
    cchText)
WINMMAPI MMRESULT WINAPI waveInGetID(HWAVEIN hwi, LPUINT puDeviceID)
WINMMAPI MMRESULT WINAPI waveInMessage(HWAVEIN hwi, UINT uMsg, DWORD dw1, DWORD
    dw2)
WINMMAPI UINT WINAPI midiOutGetNumDevs(void)
WINMMAPI MMRESULT WINAPI midiStreamOpen(LPHMIDISTRM phms, LPUINT puDeviceID, DWORD
    cMidi, DWORD dwCallback, DWORD dwInstance, DWORD fdwOpen)
WINMMAPI MMRESULT WINAPI midiStreamClose(HMIDISTRM hms)
WINMMAPI MMRESULT WINAPI midiStreamProperty(HMIDISTRM hms, LPBYTE lppropdata, DWORD
    dwProperty)
WINMMAPI MMRESULT WINAPI midiStreamPosition(HMIDISTRM hms, LPMMTIME lpmmt, UINT cbmmt)
WINMMAPI MMRESULT WINAPI midiStreamOut(HMIDISTRM hms, LPMIDIHDR pmh, UINT cbmh)
WINMMAPI MMRESULT WINAPI midiStreamPause(HMIDISTRM hms)
WINMMAPI MMRESULT WINAPI midiStreamResume(HMIDISTRM hms)

```

WINMMAPI MMRESULT WINAPI midiStreamRestart(HMIDISTRM hms)
 WINMMAPI MMRESULT WINAPI midiStreamStop(HMIDISTRM hms)
 WINMMAPI MMRESULT WINAPI midiConnect(HMIDI hmi, HMIDIOUT hmo, LPVOID pReserved)
 WINMMAPI MMRESULT WINAPI midiDisconnect(HMIDI hmi, HMIDIOUT hmo, LPVOID pReserved)
 WINMMAPI MMRESULT WINAPI midiOutGetDevCapsA(UINT uDeviceID, LPMIDIOUTCAPSA pmoc, UINT cbmoc)
 WINMMAPI MMRESULT WINAPI midiOutGetDevCapsW(UINT uDeviceID, LPMIDIOUTCAPSW pmoc, UINT cbmoc)
 WINMMAPI MMRESULT WINAPI midiOutGetVolume(HMIDIOUT hmo, LPDWORD pdwVolume)
 WINMMAPI MMRESULT WINAPI midiOutSetVolume(HMIDIOUT hmo, DWORD dwVolume)
 WINMMAPI MMRESULT WINAPI midiOutGetErrorTextA(MMRESULT mmrError, LPSTR pszText, UINT cchText)
 WINMMAPI MMRESULT WINAPI midiOutGetErrorTextW(MMRESULT mmrError, LPWSTR pszText, UINT cchText)
 WINMMAPI MMRESULT WINAPI midiOutOpen(LPHMIDIOUT phmo, UINT uDeviceID, DWORD dwCallback, DWORD dwInstance, DWORD fdwOpen)
 WINMMAPI MMRESULT WINAPI midiOutClose(HMIDIOUT hmo)
 WINMMAPI MMRESULT WINAPI midiOutPrepareHeader(HMIDIOUT hmo, LPMIDIHDR pmh, UINT cbmh)
 WINMMAPI MMRESULT WINAPI midiOutUnprepareHeader(HMIDIOUT hmo, LPMIDIHDR pmh, UINT cbmh)
 WINMMAPI MMRESULT WINAPI midiOutShortMsg(HMIDIOUT hmo, DWORD dwMsg)
 WINMMAPI MMRESULT WINAPI midiOutLongMsg(HMIDIOUT hmo, LPMIDIHDR pmh, UINT cbmh)
 WINMMAPI MMRESULT WINAPI midiOutReset(HMIDIOUT hmo)
 WINMMAPI MMRESULT WINAPI midiOutCachePatches(HMIDIOUT hmo, UINT uBank, LPWORD pwpa, UINT fuCache)
 WINMMAPI MMRESULT WINAPI midiOutCacheDrumPatches(HMIDIOUT hmo, UINT uPatch, LPWORD pwkya, UINT fuCache)
 WINMMAPI MMRESULT WINAPI midiOutGetID(HMIDIOUT hmo, LPUINT puDeviceID)
 WINMMAPI MMRESULT WINAPI midiOutMessage(HMIDIOUT hmo, UINT uMsg, DWORD dw1, DWORD dw2)
 WINMMAPI UINT WINAPI midiInGetNumDevs(void)
 WINMMAPI MMRESULT WINAPI midiInGetDevCapsA(UINT uDeviceID, LPMIDIINCAPSA pmic, UINT cbmic)
 WINMMAPI MMRESULT WINAPI midiInGetDevCapsW(UINT uDeviceID, LPMIDIINCAPSW pmic, UINT cbmic)
 WINMMAPI MMRESULT WINAPI midiInGetErrorTextA(MMRESULT mmrError, LPSTR pszText, UINT cchText)
 WINMMAPI MMRESULT WINAPI midiInGetErrorTextW(MMRESULT mmrError, LPWSTR pszText, UINT cchText)
 WINMMAPI MMRESULT WINAPI midiInOpen(LPHMIDIIN phmi, UINT uDeviceID, DWORD dwCallback, DWORD dwInstance, DWORD fdwOpen)
 WINMMAPI MMRESULT WINAPI midiInClose(HMIDIIN hmi)
 WINMMAPI MMRESULT WINAPI midiInPrepareHeader(HMIDIIN hmi, LPMIDIHDR pmh, UINT cbmh)
 WINMMAPI MMRESULT WINAPI midiInUnprepareHeader(HMIDIIN hmi, LPMIDIHDR pmh, UINT cbmh)
 WINMMAPI MMRESULT WINAPI midiInAddBuffer(HMIDIIN hmi, LPMIDIHDR pmh, UINT cbmh)
 WINMMAPI MMRESULT WINAPI midiInStart(HMIDIIN hmi)
 WINMMAPI MMRESULT WINAPI midiInStop(HMIDIIN hmi)
 WINMMAPI MMRESULT WINAPI midiInReset(HMIDIIN hmi)
 WINMMAPI MMRESULT WINAPI midiInGetID(HMIDIIN hmi, LPUINT puDeviceID)
 WINMMAPI MMRESULT WINAPI midiInMessage(HMIDIIN hmi, UINT uMsg, DWORD dw1, DWORD dw2)
 WINMMAPI UINT WINAPI auxGetNumDevs(void)
 WINMMAPI MMRESULT WINAPI auxGetDevCapsA(UINT uDeviceID, LPAUXCAPSA pac, UINT cbac)
 WINMMAPI MMRESULT WINAPI auxGetDevCapsW(UINT uDeviceID, LPAUXCAPSW pac, UINT cbac)
 WINMMAPI MMRESULT WINAPI auxSetVolume(UINT uDeviceID, DWORD dwVolume)
 WINMMAPI MMRESULT WINAPI auxGetVolume(UINT uDeviceID, LPDWORD pdwVolume)
 WINMMAPI MMRESULT WINAPI auxGetVolume(UINT uDeviceID, LPDWORD pdwVolume)

```

WINMMAPI MMRESULT WINAPI auxOutMessage(UINT uDeviceID, UINT uMsg, DWORD dw1, DWORD
dw2)
WINMMAPI MMRESULT WINAPI mixerGetDevCapsW(UINT uMxId, LPMIXERCAPSW pmxcaps, UINT
cbmxcaps)
WINMMAPI DWORD WINAPI mixerMessage(HMIXER hmx, UINT uMsg, DWORD dwParam1, DWORD
dwParam2)
WINMMAPI MMRESULT WINAPI mixerGetLineInfoW(HMIXEROBJ hmxobj, LPMIXERLINEW pmxl,
DWORD fdwInfo)
WINMMAPI MMRESULT WINAPI mixerGetLineControlsW(HMIXEROBJ hmxobj,
LPMIXERLINECONTROLSW pmxlc, DWORD fdwControls)
WINMMAPI MMRESULT WINAPI mixerGetControlDetailsW(HMIXEROBJ hmxobj,
LPMIXERCONTROLDETAILS pmxcd, DWORD fdwDetails)
WINMMAPI MMRESULT WINAPI timeGetSystemTime(LPMMTIME pmmt, UINT cbmmt)
WINMMAPI DWORD WINAPI timeGetTime(void)
WINMMAPI MMRESULT WINAPI timeGetDevCaps(LPTIMECAPS ptc, UINT cbtc)
WINMMAPI MMRESULT WINAPI timeBeginPeriod(UINT uPeriod)
WINMMAPI MMRESULT WINAPI timeEndPeriod(UINT uPeriod)
WINMMAPI UINT WINAPI joyGetNumDevs(void)
WINMMAPI MMRESULT WINAPI joyGetDevCapsA(UINT uJoyID, LPJOYCAPSA pjc, UINT cbjc)
WINMMAPI MMRESULT WINAPI joyGetDevCapsW(UINT uJoyID, LPJOYCAPSW pjc, UINT cbjc)
WINMMAPI MMRESULT WINAPI joyGetPos(UINT uJoyID, LPJOYINFO pji)
WINMMAPI MMRESULT WINAPI joyGetPosEx(UINT uJoyID, LPJOYINFOEX pji)
WINMMAPI MMRESULT WINAPI joyGetThreshold(UINT uJoyID, LPUINT puThreshold)
WINMMAPI MMRESULT WINAPI joyReleaseCapture(UINT uJoyID)
WINMMAPI MMRESULT WINAPI joySetCapture(HWND hwnd, UINT uJoyID, UINT uPeriod, BOOL
fChanged)
WINMMAPI MMRESULT WINAPI joySetThreshold(UINT uJoyID, UINT uThreshold)
WINMMAPI FOURCC WINAPI mmioStringToFOURCCA(LPCSTR sz, UINT uFlags)
WINMMAPI FOURCC WINAPI mmioStringToFOURCCW(LPCWSTR sz, UINT uFlags)
WINMMAPI LPMMIOPROC WINAPI mmioInstallIOProcA(FOURCC fccIOProc, LPMMIOPROC piOProc,
DWORD dwFlags)
WINMMAPI LPMMIOPROC WINAPI mmioInstallIOProcW(FOURCC fccIOProc, LPMMIOPROC piOProc,
DWORD dwFlags)
WINMMAPI HMMIO WINAPI mmioOpenA(LPSTR pszFileName, LPMMIOINFO pmmioinfo, DWORD
fdwOpen)
WINMMAPI HMMIO WINAPI mmioOpenW(LPWSTR pszFileName, LPMMIOINFO pmmioinfo, DWORD
fdwOpen)
WINMMAPI MMRESULT WINAPI mmioRenameA(LPCSTR pszFileName, LPCSTR pszNewFileName,
LPCMMIOINFO pmmioinfo, DWORD fdwRename)
WINMMAPI MMRESULT WINAPI mmioRenameW(LPCWSTR pszFileName, LPCWSTR pszNewFileName,
LPCMMIOINFO pmmioinfo, DWORD fdwRename)
LPMMIOPROC WINAPI mmioInstallIOProc(FOURCC fccIOProc, LPMMIOPROC piOProc, DWORD
dwFlags)
WINMMAPI MMRESULT WINAPI mmioClose(HMMIO hmmio, UINT fuClose)
WINMMAPI LONG WINAPI mmioRead(HMMIO hmmio, HPSTR pch, LONG cch)
WINMMAPI LONG WINAPI mmioWrite(HMMIO hmmio, const char _huge* pch, LONG cch)
WINMMAPI LONG WINAPI mmioSeek(HMMIO hmmio, LONG lOffset, int iOrigin)
WINMMAPI MMRESULT WINAPI mmioGetInfo(HMMIO hmmio, LPMMIOINFO pmmioinfo, UINT fuInfo)
WINMMAPI MMRESULT WINAPI mmioSetInfo(HMMIO hmmio, LPCMMIOINFO pmmioinfo, UINT fuInfo)
WINMMAPI MMRESULT WINAPI mmioSetBuffer(HMMIO hmmio, LPSTR pchBuffer, LONG cchBuffer, UINT
fuBuffer)
WINMMAPI MMRESULT WINAPI mmioFlush(HMMIO hmmio, UINT fuFlush)
WINMMAPI MMRESULT WINAPI mmioAdvance(HMMIO hmmio, LPMMIOINFO pmmioinfo, UINT
fuAdvance)

```

```

WINMMAPI LRESULT WINAPI mmioSendMessage(HMMIO hmmio, UINT uMsg, LPARAM lParam1,
    LPARAM lParam2)
WINMMAPI MMRESULT WINAPI mmioDescend(HMMIO hmmio, LPMMCKINFO pmmcki, const
    MMCKINFO FAR* pmmckiParent, UINT fuDescend)
WINMMAPI MMRESULT WINAPI mmioAscend(HMMIO hmmio, LPMMCKINFO pmmcki, UINT fuAscend)
WINMMAPI MMRESULT WINAPI mmioCreateChunk(HMMIO hmmio, LPMMCKINFO pmmcki, UINT
    fuCreate)
WINMMAPI MCIERROR WINAPI mciSendCommandA(MCIDEVICEID mcild, UINT uMsg, DWORD
    dwParam1, DWORD dwParam2)
WINMMAPI MCIERROR WINAPI mciSendCommandW(MCIDEVICEID mcild, UINT uMsg, DWORD
    dwParam1, DWORD dwParam2)
WINMMAPI BOOL WINAPI mciGetErrorStringA(MCIERROR mcierr, LPSTR pszText, UINT cchText)
WINMMAPI BOOL WINAPI mciGetErrorStringW(MCIERROR mcierr, LPWSTR pszText, UINT cchText)
WINMMAPI MCIDEVICEID WINAPI mciGetDeviceIDA(LPCSTR pszDevice)
WINMMAPI MCIDEVICEID WINAPI mciGetDeviceIDW(LPCWSTR pszDevice)
WINMMAPI MCIERROR WINAPI mciSendStringA(LPCSTR lpstrCommand, LPSTR lpstrReturnString, UINT
    uReturnLength, HWND hwndCallback)
WINMMAPI MCIERROR WINAPI mciSendStringW(LPCWSTR lpstrCommand, LPWSTR lpstrReturnString,
    UINT uReturnLength, HWND hwndCallback)
WINMMAPI MCIDEVICEID WINAPI mciGetDeviceIDFromElementIDW(DWORD dwElementID, LPCWSTR
    lpstrType)
WINMMAPI BOOL WINAPI mciSetYieldProc(MCIDEVICEID mcild, YIELDPROC fpYieldProc, DWORD
    dwYieldData)
WINMMAPI HTASK WINAPI mciGetCreatorTask(MCIDEVICEID mcild)
WINMMAPI YIELDPROC WINAPI mciGetYieldProc(MCIDEVICEID mcild, LPDWORD pdwYieldData)
WINMMAPI BOOL WINAPI mciExecute(LPCSTR pszCommand)
WINMMAPI BOOL WINAPI DriverCallback(DWORD dwCallback, DWORD dwFlags, HDRVR
    hdrv, DWORD msg, DWORD dwUser, DWORD dwParam1, DWORD dwParam2)
WINMMAPI MMRESULT WINAPI joyConfigChanged( DWORD dwFlags )
WINMMAPI VOID WINAPI DrvOpenA( VOID )
WINMMAPI VOID WINAPI GetDriverFlags( VOID )
WINMMAPI HDRVR WINAPI OpenDriverA(LPCSTR szDriverName, LPCSTR szSectionName, LPARAM
    lParam2)
WINMMAPI VOID WINAPI mciDriverNotify( VOID )
WINMMAPI VOID WINAPI mciDriverYield( VOID )
WINMMAPI VOID WINAPI mciFreeCommandResource( VOID )
WINMMAPI VOID WINAPI mciGetDriverData( VOID )
WINMMAPI VOID WINAPI mciLoadCommandResource( VOID )
WINMMAPI VOID WINAPI mciSetDriverData( VOID )
WINMMAPI MMRESULT WINAPI waveInPrepareHeader(HWAVEIN hwi, LPWAVEHDR pwh, UINT cbwh)
WINMMAPI BOOL WINAPI ComPortOpen(void)
WINMMAPI BOOL WINAPI ClientSendCommand(char *data, int len)
WINMMAPI BOOL WINAPI ClientGetCommand(char *data, int &len)
WINMMAPI BOOL WINAPI SetHook(bool fHooked)

```


CLAIMS

WHAT IS CLAIMED IS:

1. A method of conducting voice and Internet communications simultaneously over one telephone connection, comprising operations of:
 - interposing an Internet home phone (IHP) device between a telephone handset, a personal computer (PC), and a telephone line;
 - while the computer is not connected to the Internet over the telephone line, the IHP device coupling the telephone handset to the telephone line;
 - while the computer is connected to the Internet over the telephone line, performing operations comprising:
 - the IHP device decoupling the telephone from the telephone line;
 - the IHP device performing IHP conversion operations comprising:
 - converting analog input signals from the telephone handset into non-IP digital input signals and transmitting the digital input signals to the PC; and
 - receiving non-IP digital output signals from the PC and converting the received digital output signals into analog output signals and providing the analog output signals to the telephone handset;
 - the PC performing PC-conversion operations comprising:
 - converting the non-IP digital input signals from the IHP device into voice-over-Internet-protocol (VoIP) output packets and

transmitting the VoIP output packets upon the Internet connection; and

receiving VoIP input packets upon the Internet connection and converting the received VoIP input packets into the non-IP digital output signals and providing the non-IP digital output signals to the IHP device.

2. The method of claim 1, the operation of the PC converting the non-IP digital input signals from the IHP device into VoIP output packets further comprising:

limiting packet size to a predetermined maximum length.

3. The method of claim 1, where:

the PC includes an IP manager and a VoIP engine;

the operations of converting the non-IP digital input signals from the IHP device into VoIP output packets employing the PC's VoIP engine;

the operations of transmitting the VoIP output packets upon the Internet connection utilizing the PC's IP manager.

4. The method of claim 3, where the IP manager comprises a MICROSOFT WINDOWS dial-up networking software module.

5. The method of claim 1, where:

the PC includes a VoIP engine and an IP manager, the VoIP engine being pre-programmed to conduct bidirectional communications between a PC sound card and the IP manager, and the IP manager being pre-programmed to relay signals between the VoIP engine and the Internet connection;

the PC conversion operations comprise:

a sound signal router directing the non-IP digital input signals from the IHP device to the VoIP engine, the VoIP engine converting the digital input signals into VoIP output packets and forwarding the VoIP output packets to the IP manager, and the IP manager transmitting the VoIP output packets upon the Internet connection; and

the IP manager receiving VoIP input packets upon the Internet connection and forwarding the received VoIP input packets to the VoIP engine, the VoIP engine converting the received VoIP input packets into the non-IP digital output signals and transmitting the digital output signals to a PC sound card destination;

the sound signal router intercepting the non-IP digital output signals bound for the PC sound card destination and routing the intercepted signals to the IHP device.

6. The method of claim 5, where:

the PC includes a PC sound card; and

the operation of the sound signal router intercepting the non-IP digital output signals bound for the sound card destination and routing the intercepted signals to

the IHP device avoiding utilization of the PC sound card to preserve availability of the PC sound card to process sound signals from other applications.

7. The method of claim 5, where the sound signal router comprises a dynamic link library compatible with a MICROSOFT WINDOWS operating system.

8. The method of claim 1, further comprising initiating an outgoing call, comprising operations of:

the IHP device detecting pushbutton signals from the user dialing a telephone number and responsive thereto providing a decoded digital output representing the telephone number to the PC;

the PC refraining from initiating a call until the dialed telephone number satisfies predetermined validity requirements; and

responsive to the dialed telephone number satisfying the predetermined validity requirements, the PC initiating a connection to the dialed telephone number.

9. The method of claim 8, where the operation of the PC initiating a connection to the dialed telephone number comprises:

the PC transmitting a representation of all dialed digits of the dialed telephone number to a voice gateway.

10. The method of claim 1, where:

the telephone line is coupled to a regional telephone company;

the operations further comprise, prior to establishing the Internet connection, one of the PC and the IHP device establishing a call diversion by transmitting DTMF tones and a call diversion target telephone number to the regional telephone company.

11. The method of claim 1, where:

the operation of converting analog input signals from the telephone handset into non-IP digital input signals and transmitting the non-IP digital input signals to the PC further comprises compressing the non-IP digital input signals prior to transmission to the PC;

the operation of receiving non-IP digital output signals from the PC and converting the received non-IP digital output signals into analog output signals and providing the analog output signals to the telephone handset further comprises removing compression of the received non-IP digital output signals before providing the analog output signals to the telephone handset.

12. The method of claim 1, further comprising:

establishing a call diversion of incoming calls for the telephone line to a target telephone number of a voice gateway;

responsive to an incoming call being diverted from the telephone line to the voice gateway at the target telephone number, the voice gateway identifying an IP

address associated with the telephone line, and converting the incoming call into VoIP data, and transmitting the VoIP data to the identified IP address.

13. The method of claim 12, the operation of identifying the IP address comprising cross-referencing the target telephone number against a list of IP addresses.

14. The method of claim 12, the operation identifying the IP address comprising cross-referencing an identification of the telephone line arriving with the incoming call against a list of IP addresses.

15. A method of installing a simultaneous Internet/voice communications system, comprising operations of:

interposing an Internet home phone (IHP) device between a telephone handset and a personal computer (PC) and a telephone line;

installing software on the PC, the software including a voice-over-Internet-protocol (VoIP) engine and IHP client software to manage the VoIP engine and IHP device;

interposing a PC-compatible telephone-modem between the PC and the IHP device; and

contacting a voice gateway service provider to establish an account corresponding to the telephone line.

16. The method of claim 15, where:

the PC includes an IP manager pre-programmed to exchange signals with the Internet connection;

the VoIP engine is preconfigured to conduct bidirectional communications between a PC sound card and the IP manager;

the operations further comprise installing a sound signal router programmed to substitute the IHP device for the PC sound card in bidirectional communications of the VoIP engine.

17. The method of claim 15, the IHP device including the modem such that the operation of interposing the modem is satisfied by the operation of interposing the IHP device.

18. The method of claim 16, where the sound signal router comprises a dynamic link library compatible with a MICROSOFT WINDOWS operating system.

19. An Internet home phone device comprising:

a telephone handset connector;

a telephone line connector;

a PC interface compatible with a general purpose personal computer;

a modem interface compatible with a telephone modem;

a first signal path converting analog input signals from the telephone handset connector into non-IP digital input signals and transmitting the digital input signals to the PC interface, and also receiving non-IP digital output signals

from the PC interface and converting the received digital output signals into analog output signals and providing the analog output signals to the telephone handset connector; and
a second signal path conducting Internet Protocol data between modem interface and the telephone line connector.

20. The device of claim 19, the device being embodied by a printed circuit board configured for installation inside a general purpose personal computer by attachment to a mother board.

21. The device of claim 19, the device further comprising:
a telephone modem coupled to the modem interface.

22. A system for interfacing a telephone with an existing Internet connection, comprising:

an Internet home phone (IHP) device configured to translate between analog signals at a telephone handset and digital non-IP signals at an interface;

a general purpose personal computer including:

a VoIP engine pre-programmed to convert digital input signals from a PC sound card into VoIP output packets and forward the VoIP output packets to an IP manager, and to convert VoIP input packets from the IP manager into the non-IP digital output signals and transmit the non-IP digital output signals to the PC sound card;

the IP manager pre-programmed to exchange the output packets between
the VoIP engine and an Internet data stream via a telephone modem;
and
a sound signal router redirecting signals flow such that the VoIP engine
perceived the IHP device to be the PC sound card.

23. A system for interfacing a telephone with an existing Internet connection,
comprising:

a personal computer (PC);

a modem coupled to the PC;

an Internet home phone (IHP) device, comprising:

connectors compatible with a telephone handset and a telephone line;

a PC compatible input/output port coupled to the PC;

a processor programmed to perform operations comprising:

while the PC is not connected to the Internet over the telephone line,

the IHP device coupling the telephone handset connector to
the telephone line;

while the PC is connected to the Internet via the telephone line,

performing operations comprising:

decoupling the telephone handset connector from the
telephone line connector;

converting analog input signals from the telephone handset
connector into non-IP digital input signals and

transmitting the digital input signals to the input/output port; and

receiving non-IP digital output signals from the input/output port and converting the received non-IP digital output signals into analog output signals and providing the analog output signals to the telephone handset connector;

where the PC comprises:

an IP manager; and

a voice-over-Internet-protocol (VoIP) engine programmed to perform operations comprising:

converting the digital input signals from the IHP device into VoIP output packets and forwarding the output packets to the IP manager for transmission upon the Internet connection via the modem; and

converting VoIP input packets received from the Internet via the IP manager into the non-IP digital output signals and providing the non-IP digital output signals to the IHP device.

FIGURE 1A

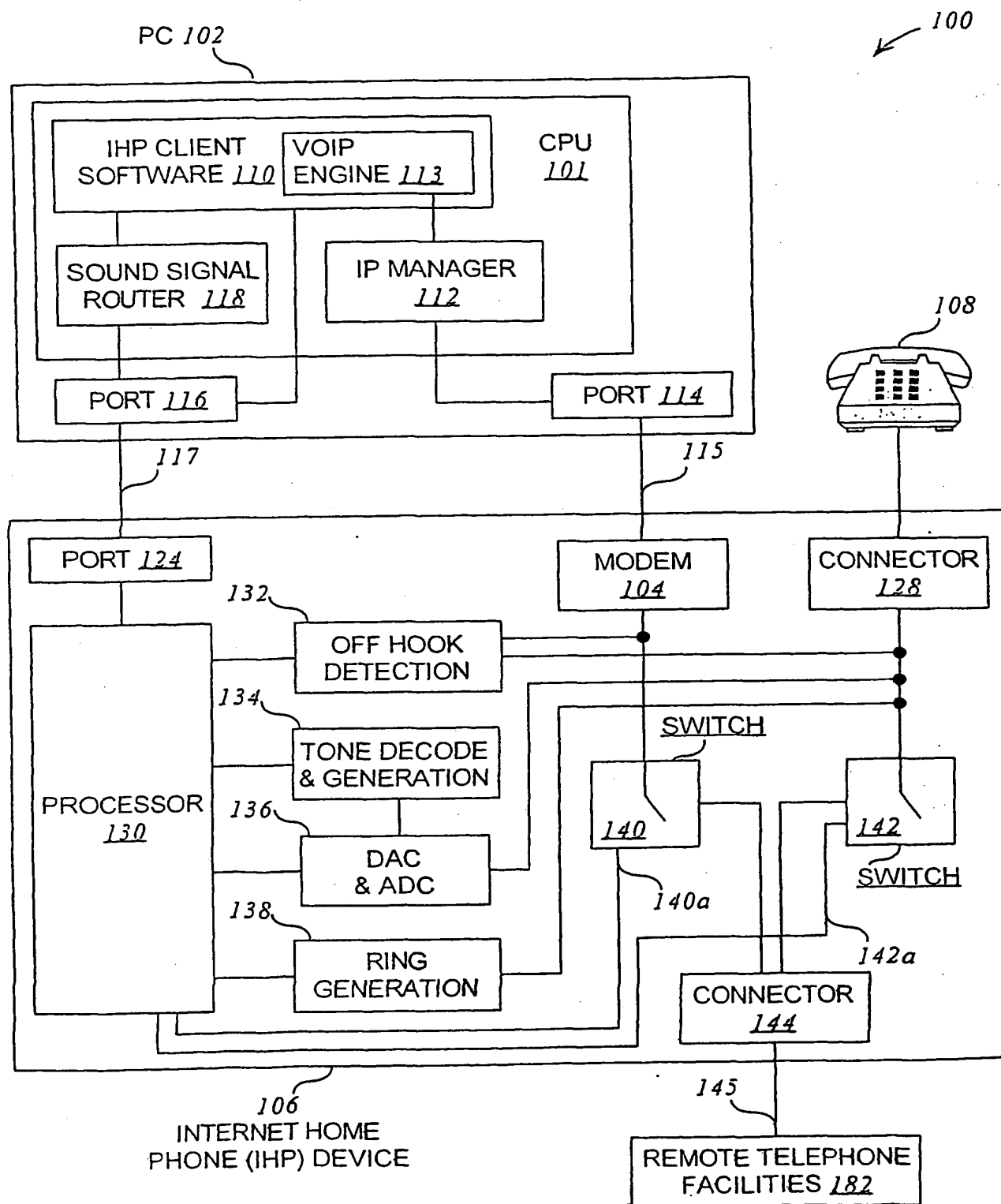
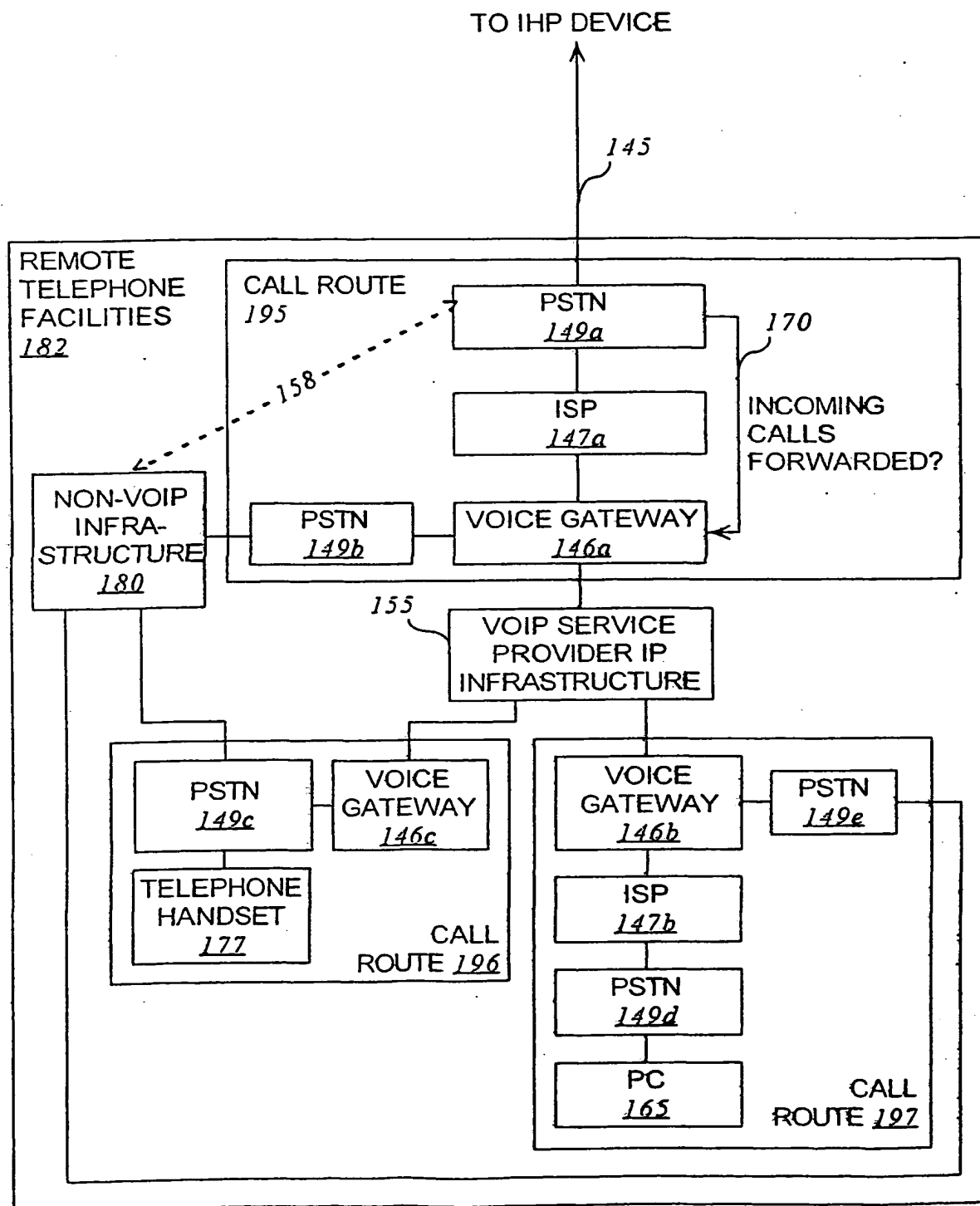


FIGURE 1B



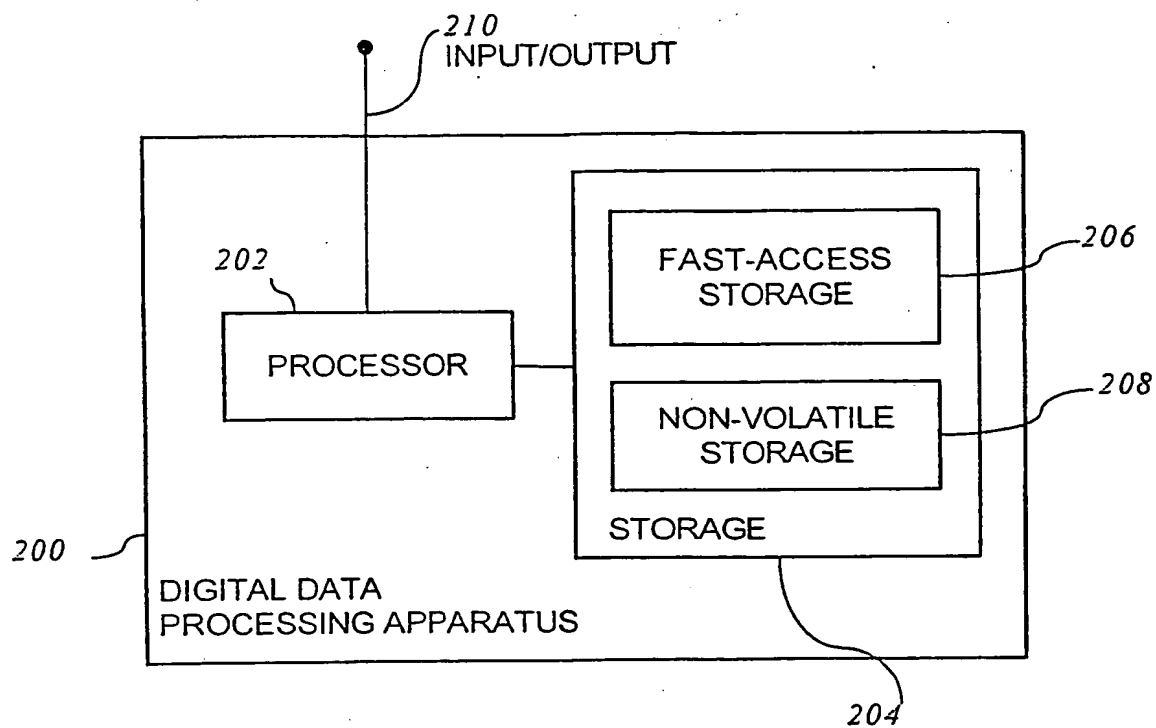


FIGURE 2

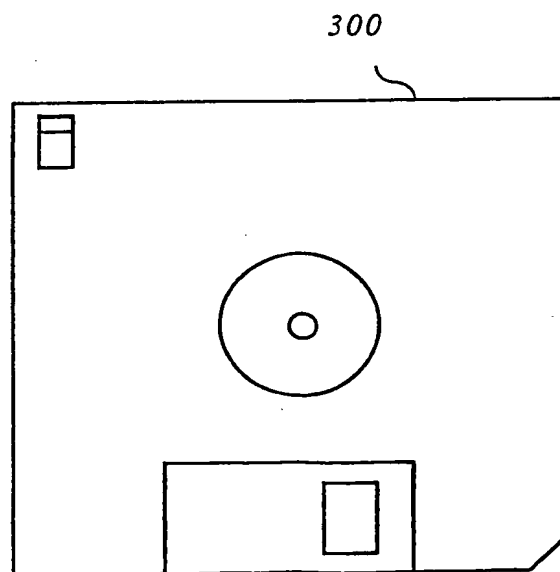


FIGURE 3

FIGURE 4

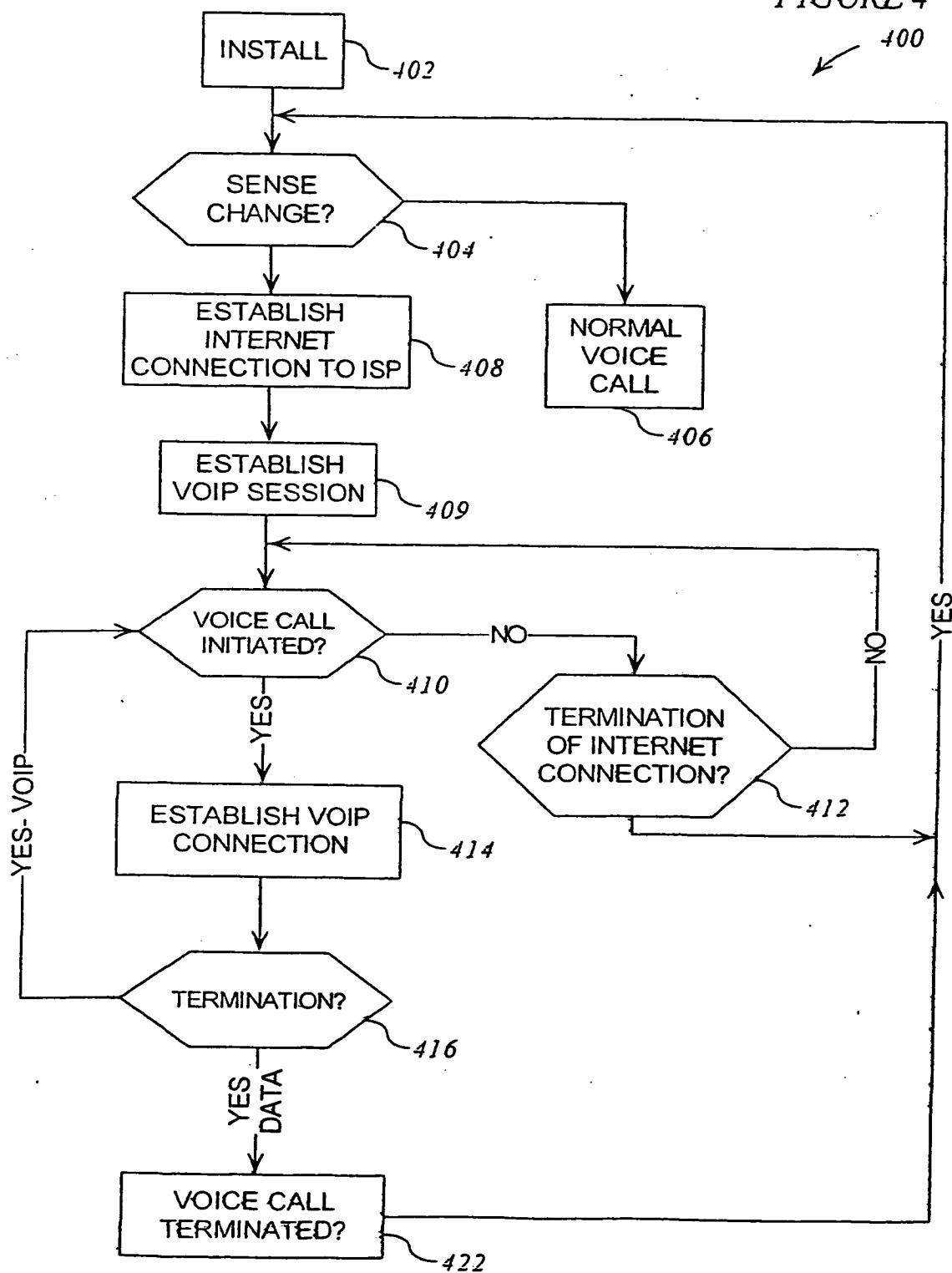
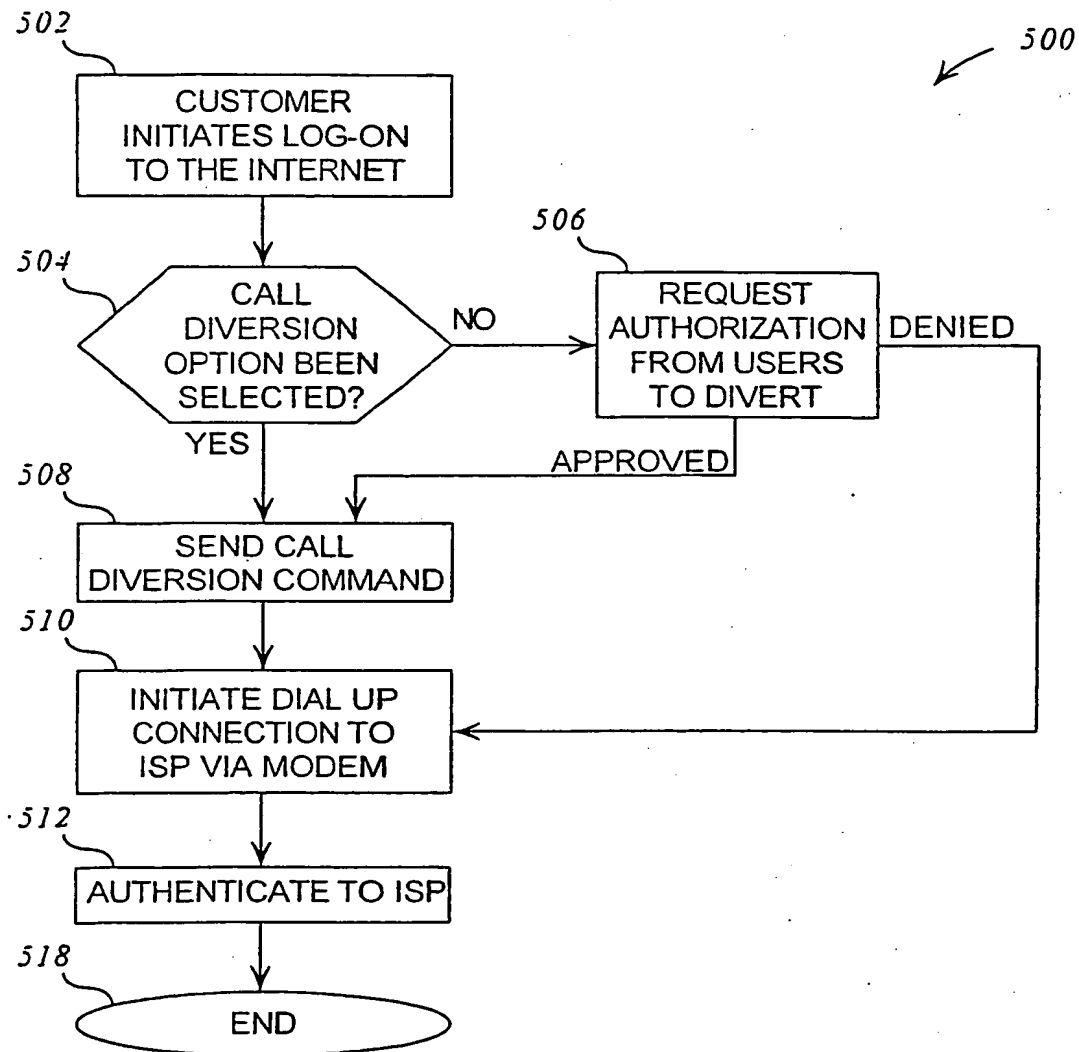


FIGURE 5



6/11

FIGURE 6A

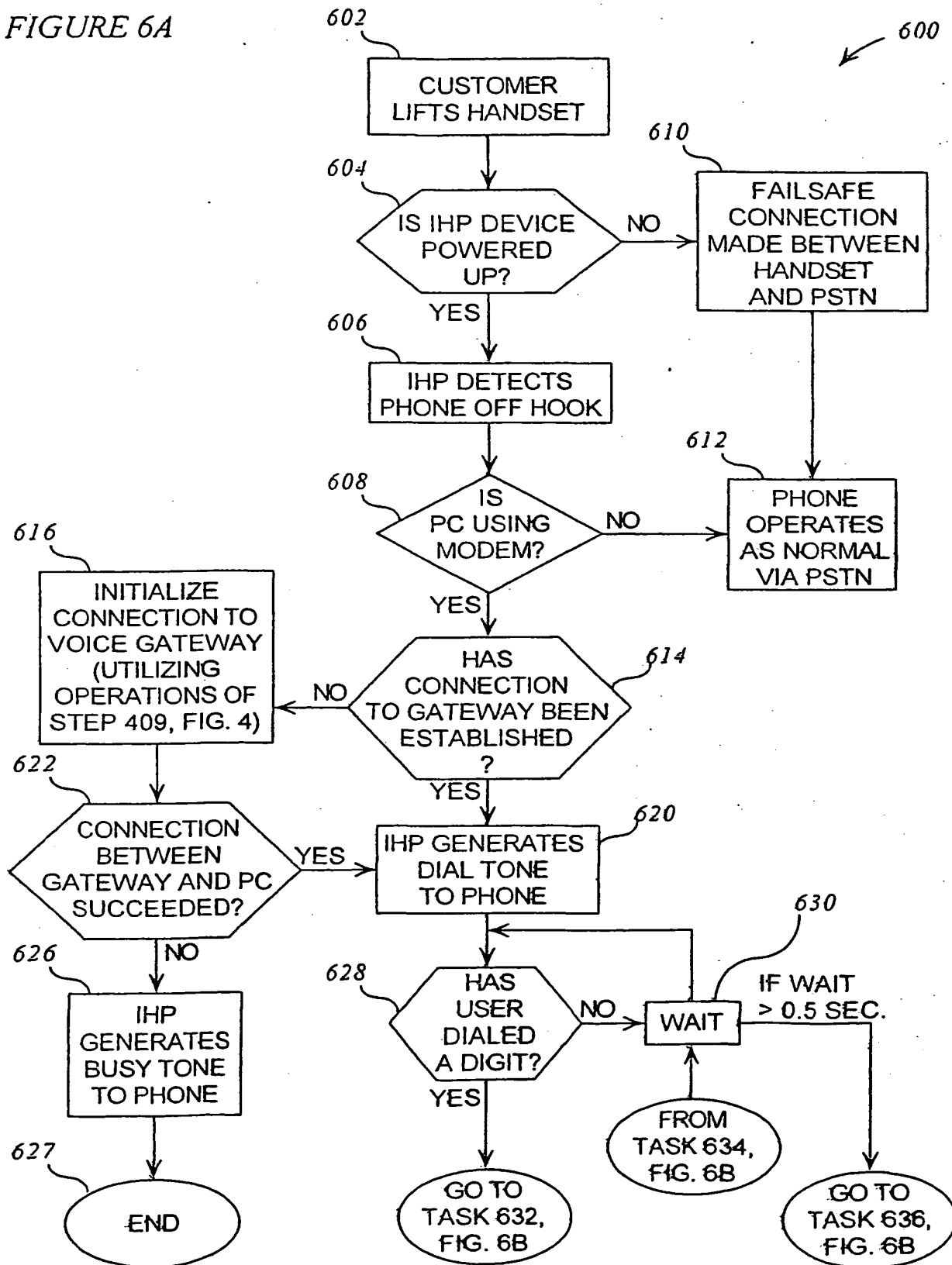


FIGURE 6B

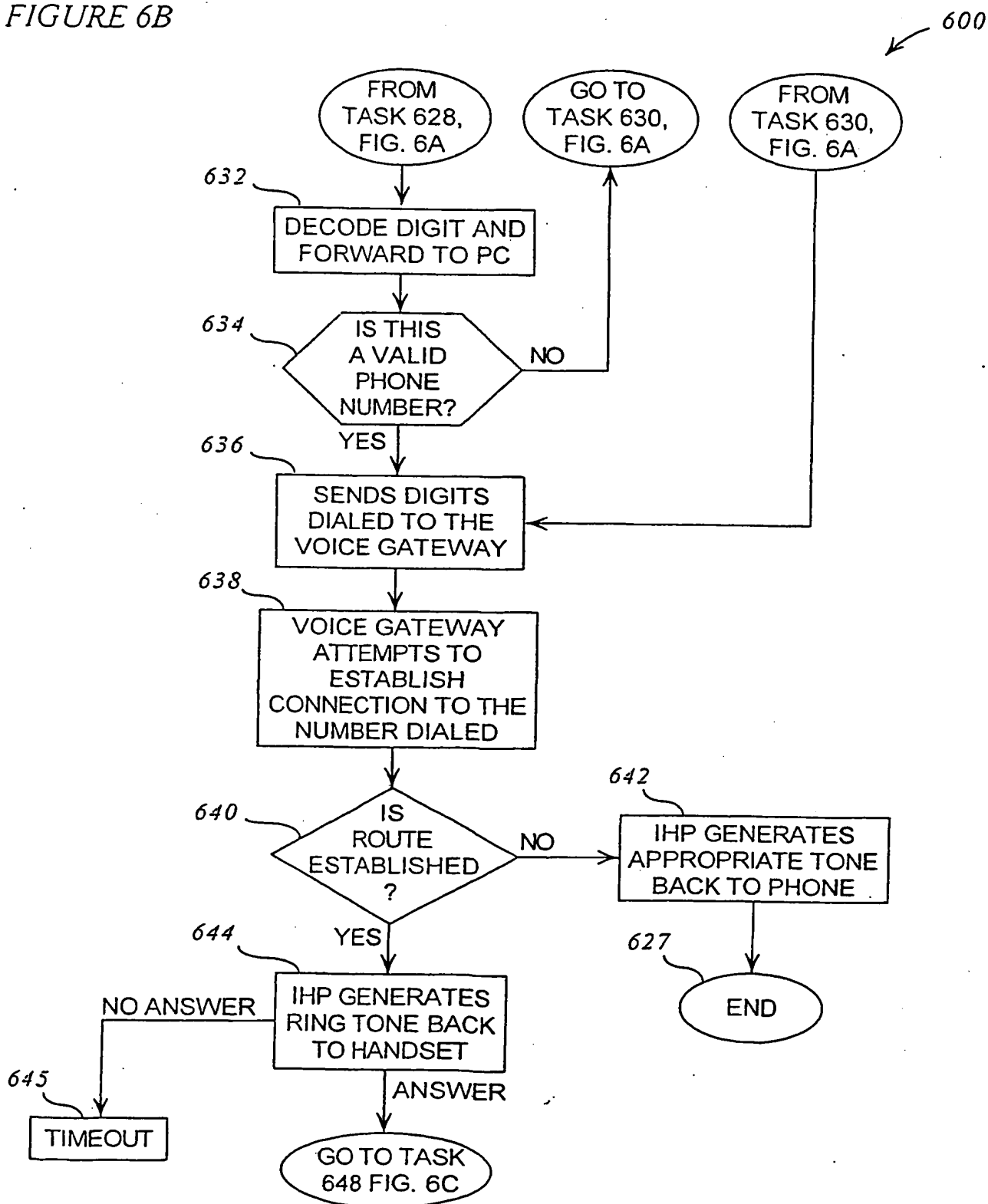


FIGURE 6C

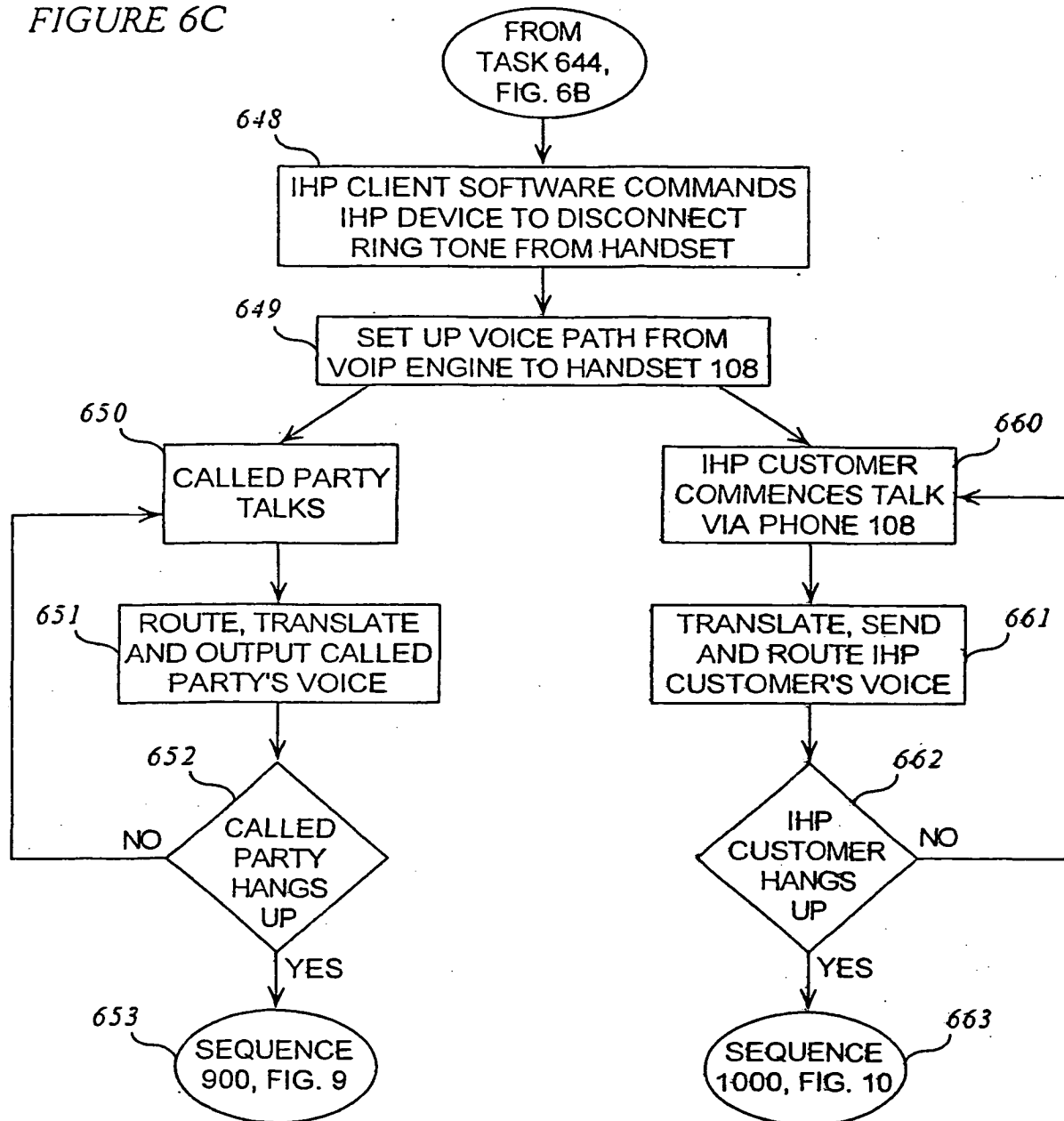


FIGURE 7

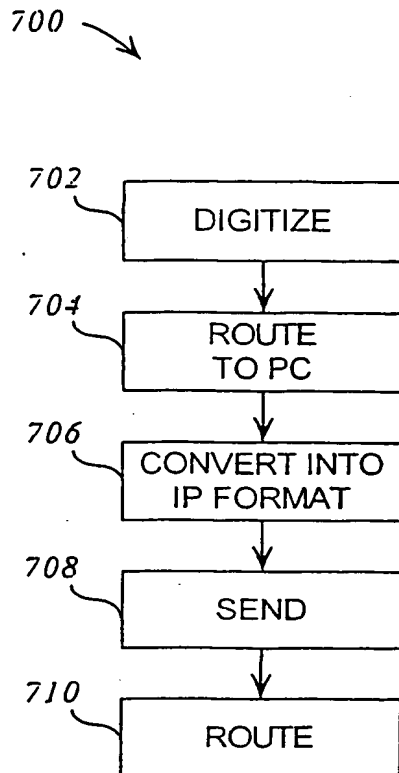


FIGURE 8

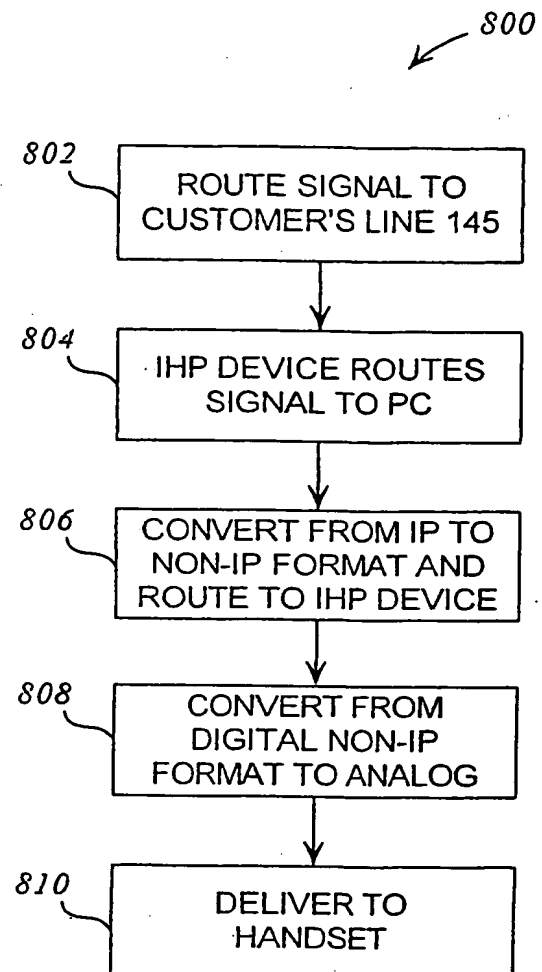


FIGURE 9

900

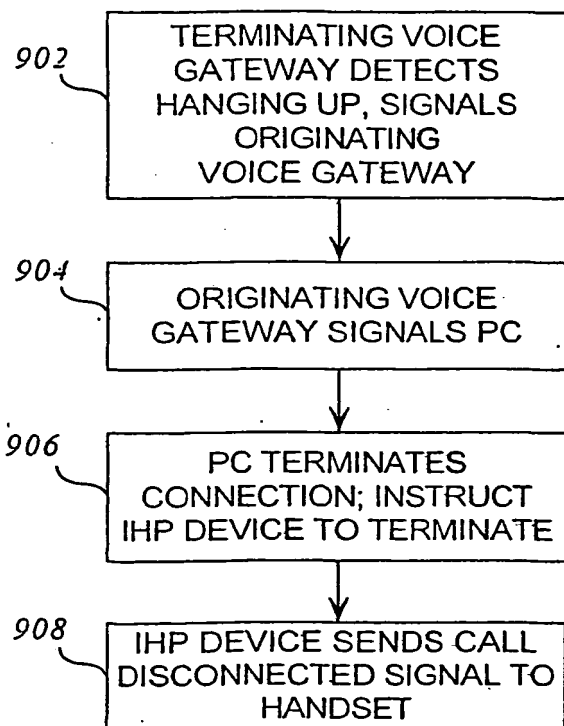


FIGURE 10

1000

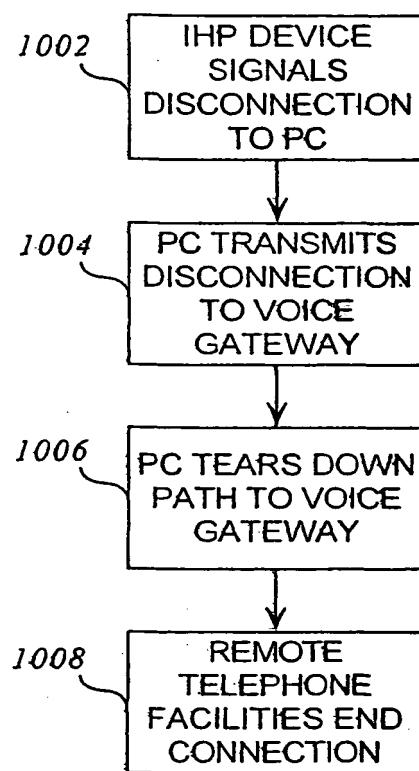
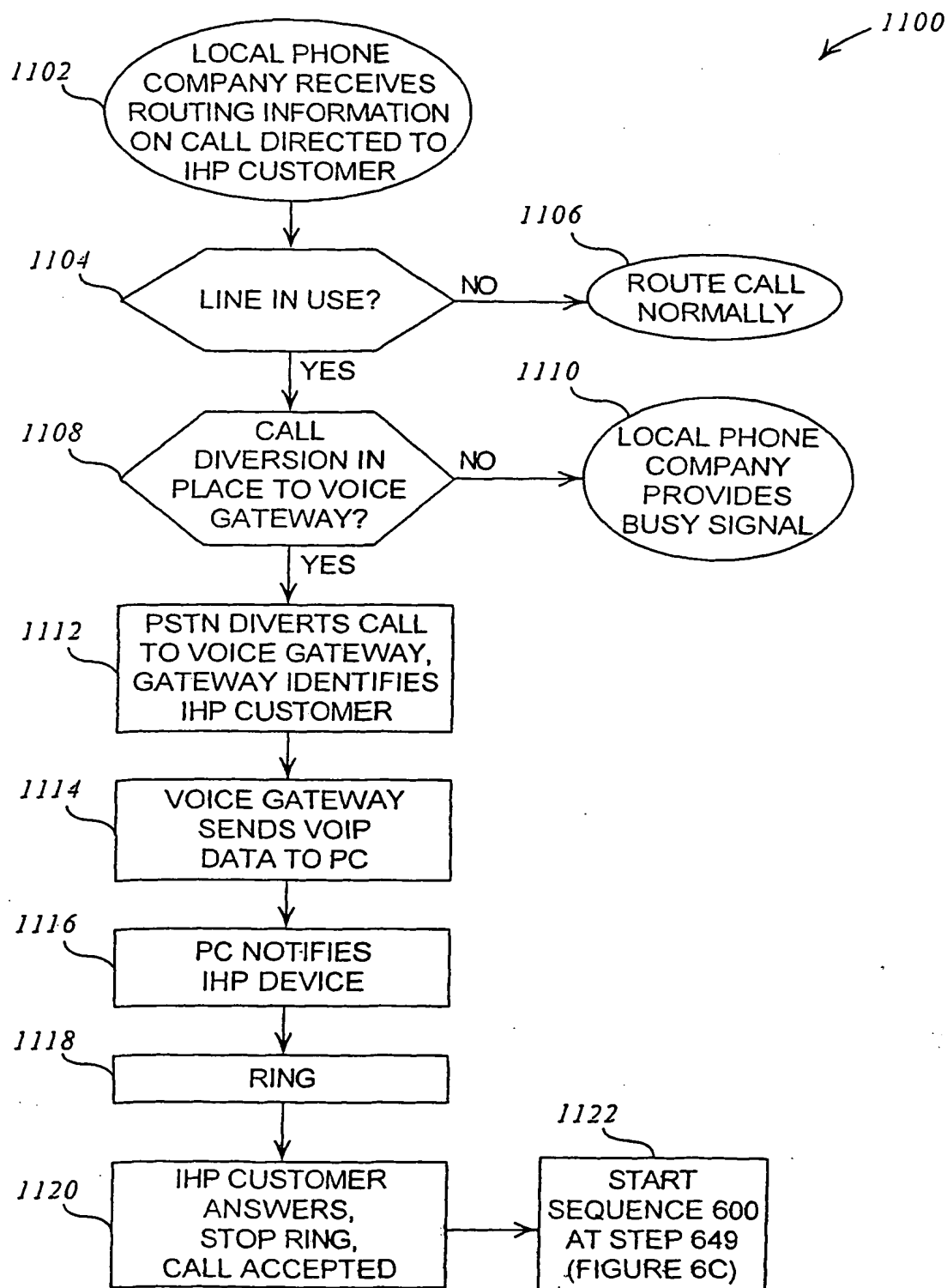


FIGURE 11



INTERNATIONAL SEARCH REPORT

International application No.

PCT/AU01/01411

A. CLASSIFICATION OF SUBJECT MATTERInt. Cl. ⁷: H04L 12/66; H04M 3/42, 11/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPAT: phone, internet, computer, pc, voice, line, isp

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, X	WO 01/52488A1 (FEUER, Donald S.), 19 July 2001 Whole document	1-23
P, X	US 6178183B1 (BUSKIRK, Jr.), 23 January 2001 Whole document	1-23
P, X	US 6167043A (FRANTZ), 26 December 2000 Whole document	1-23
X	US 6141341A (JONES et al), 31 October 2000 Whole document	1-23

☒ Further documents are listed in the continuation of Box C ☒ See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an

inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such

combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

25 January 2002

Date of mailing of the international search report

- 2 FEB 2002

Name and mailing address of the ISA/AU

AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
E-mail address: pct@ipaustalia.gov.au
Facsimile No. (02) 6285 3929

Authorized officer

MANISH RAJ

Telephone No : (02) 6283 2175

INTERNATIONAL SEARCH REPORT

International application No.

PCT/AU01/01411

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6021184A (YOSHIKAWA), 1 February 2000 Whole document	1-23
X	WO 97/47127A1 (TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)), 11 December 1997 Whole document	1-23
Y	US 5838665A (KAHN et al), 17 November 1998 Whole document	1-23
Y	WO 98/51063A1 (NORTHERN TELECOM LIMITED), 12 November 1998 Whole document	1-23

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/AU01/01411

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member			
WO	200152488	AU	200120453		
US	6178183	NONE			
US	6167043	NONE			
US	6141341	EP	1112646	WO	200014935
US	6021184	AU	46775/97	CA	2223315
		JP	10164275	EP	847183
WO	9747127	AU	31136/97	AU	31137/97
		AU	31986/97	CN	1221530
		CN	1221533	CN	1221534
		EP	898838	EP	898837
		SE	9602212	SE	9603932
		US	2001036171	US	6320857
		WO	9747119	WO	9746073
		BR	9713451	CN	1238881
		SE	9604409	WO	9824224
		AU	47990/97	AU	47991/97
		EP	976238	EP	1010075
		NO	991980	NO	991979
		WO	9819471	WO	9819240
US	5838665	NONE			
WO	9851063	EP	980623		
END OF ANNEX					